

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені
адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
(повне найменування інституту, назва факультету)

Програмного забезпечення автоматизованих систем
(повна назва кафедри)

Пояснювальна записка

до кваліфікаційної (магістерської) роботи

за темою Удосконалення моделі якості програмного забезпечення
діагностики двигунів та розробка програми для її реалізації

Виконав: студент 6 курсу, 6151м групи
спеціальності

121 «Інженерія програмного забезпечення»

_____ Алексєєв Д.Ю.

Керівник МР _____ Суслов С.В.

Рецензент МР _____ Приходько С.Б.

Завідувач кафедри _____ Приходько С.Б.

Миколаїв 2020р.

Міністерство освіти і науки України
Національний університет кораблебудування
імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
 Кафедра програмного забезпечення автоматизованих систем

Освітній ступень Магістр

Галузь 12 «Інформаційні технології»

(шифр і назва)

Спеціальність 121 «Інженерія програмного забезпечення»

(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри

“ 26 ” 10 2020 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ (МАГІСТЕРСЬКУ) РОБОТУ СТУДЕНТУ

Алексєєв ДанилоЮрійович

(прізвище, ім'я, по батькові)

1. Тема магістерської роботи Удосконалення моделі якості програмного забезпечення
діагностики двигунів та розробка програми для її реалізації

керівник роботи Суслов Сергій Віталійович, канд. техн. наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ 2 ” жовтня 2019 року №1124-уч

2. Строк подання студентом роботи 01.12.2020 року

3. Вихідні дані до роботи Документація з програмного забезпечення діагностики
двигунів. Стандарти якості програмного забезпечення

4. Зміст магістерської роботи (МР):

- Титульний аркуш, завдання на кваліфікаційну (магістерську) роботу, реферат (українською, англійською), зміст, перелік умовних позначень, символів, одиниць та термінів (за необхідності).
- Вступ (Актуальність теми. Зв'язок роботи з науковими програмами, планами, темами. Мета і завдання дослідження. Об'єкт дослідження. Предмет дослідження. Методи дослідження. Наукова новизна одержаних результатів. Практичне значення одержаних результатів. Особистий внесок здобувача. Апробація результатів досліджень. Публікації.)
- Огляд літератури за темою, обґрунтування необхідності проведення досліджень за обраною темою, вибір напрямків досліджень, мета дослідження, основні задачі дослідження
- Викладення результатів власних досліджень з висвітленням того нового, що пропонується
- Проект програмного забезпечення
- Результати досліджень та розробки проекту програмного забезпечення
- Організаційно-економічний розділ

• Розділи з охорони праці та охорони навколишнього середовища

- Висновки
 - Список використаних джерел
 - Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма і методика випробувань програмного забезпечення)
5. Перелік графічного матеріалу

6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 1 жовтня 2019р

КАЛЕНДАРНИЙ ПЛАН

Назва етапів кваліфікаційної (магістерської) роботи (МР)	Термін виконання	Примітка
1. Підготовка вступної частини МР	18.10.2020	18.10.2020
2. Підготовка розділу (ів) МР з огляду літератури за темою, обґрунтування необхідності проведення досліджень за обраною темою, вибір напрямів досліджень	19.10.2020	19.10.2020
3. Підготовка розділу (ів) МР з результатів власних досліджень	22.10.2020	22.10.2020
4. Підготовка розділу МР з проекту програмного забезпечення	16.11.2020	16.11.2020
5. Підготовка організаційно-економічного розділу	18.11.2020	18.11.2020
6. Підготовка розділу з охорони праці	20.11.2020	20.11.2020
7. Підготовка розділу з охорони навколишнього середовища	23.11.2020	23.11.2020
8. Підготовка розділу МР – Висновки	25.11.2020	25.11.2020
9. Оформлення списку використаних джерел	27.11.2020	27.11.2020
10. Оформлення додатків	30.11.2020	30.11.2020
11. Підготовка презентації МР та доповіді	01.12.2020	01.12.2020
12. Подання МР на попередній захист	01.12.2020	01.12.2020
13. Подання МР рецензенту	11.12.2020	11.12.2020
14. Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, разом з заявою щодо самостійності виконання роботи та ідентичності друкованої та електронної версії роботи	14.12.2020	14.12.2020
15. Подання на кафедру ПЗАС електронних версії наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК від 19.05.2020 р. за №287-уч); презентації доповіді	18.12.2020	18.12.2020
16. Подання на кафедру ПЗАС письмового відгуку та рецензії на кваліфікаційну роботу	18.12.2020	18.12.2020

Студент

_____ (підпис)

Алексеев Д.Ю.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Сулов С.В.

_____ (прізвище та ініціали)

Реферат до магістерської роботи

РЕФЕРАТ

Алексєєв Данило Юрійович

«Удосконалення моделі якості програмного забезпечення діагностики
двигунів та розробка програми для її реалізації»

Магістерська робота на здобуття освітнього рівня магістра зі спеціальності 121 – «Інженерія програмного забезпечення». Національний університет кораблебудування імені адмірала Макарова. Миколаїв, 2020 р.

Обсяг роботи: 78 стор., 10 табл., 9 рис., 30 використані джерела, 5 додатків.

Актуальність теми. Сучасний кваліфікований фахівець діагност не може обійтись без навичок володіння програмним забезпеченням для діагностики двигунів. Ці програмні продукти є достатньо складними і тому витратними у придбанні і впровадженні. Застосування моделі якості в розробці програмного забезпечення діагностики двигунів дозволяє оптимізувати цикл розробки програмного забезпечення та вигоди від його використання.

Мета і завдання дослідження. Покращення якісних характеристик програмного забезпечення для діагностики двигунів та їх реалізації.

Об'єктом дослідження є програмне забезпечення для діагностики двигунів.

Предметом дослідження є якість програмного забезпечення для діагностики двигунів

Наукова новизна одержаних результатів полягає в удосконаленні існуючої моделі якості шляхом урахування особливостей програмного забезпечення для діагностики двигунів.

Практичне значення одержаних результатів. Результатом роботи було отримання програмного забезпечення діагностики двигунів покращеного за допомогою удосконаленої моделі якості.

Ключові слова. Діагностики двигунів; моделі якості; покращення якісних характеристик; двигун; якість; програмне забезпечення діагностики.

Summary of the master's work

Alekseev Danilo Yuriyovych

"Improving the quality model of engine diagnostics software and developing a program for its implementation"

Master's Degree in Master's Degree in Specialty 121 - Software Engineering.
Admiral Makarov National University of Shipbuilding. Mykolaiv, 2020

Scope of work: 78 pages, 10 tables, 9 figures, 30 sources used, 5 applications.

Actuality of theme. A modern qualified diagnostician cannot do without the skills of owning software for engine diagnostics. These software products are quite complex and therefore costly to purchase and implement. The application of the quality model in the development of engine diagnostics software allows to optimize the software development cycle and the benefits of its use.

The purpose and objectives of the study. Improving the quality characteristics of software for engine diagnostics and their implementation.

The object of research is software for engine diagnostics.

The subject of the study is the quality of software for engine diagnostics

The scientific novelty of the obtained results is to improve the existing quality model by taking into account the features of software for engine diagnostics.

The practical significance of the obtained results. The result was a quality model for engine diagnostics software.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	9
ВСТУП.....	10
1 ДІАГНОСТИКА ДВИГУНІВ: СУЧАСНІ ТЕХНОЛОГІЇ.....	12
1.1 Програмне забезпечення діагностики двигунів.....	12
1.2 Існуючі програмні рішення для діагностики двигунів	13
1.3 Висновки до розділу 1.....	14
2 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ЯКОСТІ ПРОГРАМНОГО ЗАБЕПЕЧЕННЯ.....	15
2.1 Характеристика якості програмного забезпечення діагностики двигунів.....	15
2.2 Існуючі моделі якості програмного забезпечення.....	16
2.3 Шляхи удосконалення моделі якості програмного забезпечення діагностики.....	17
2.4 Висновки до розділу 2.....	18
3 УДОСКОНАЛЕННЯ МОДЕЛІ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДІАГНОСТИКИ ДВИГУНІВ.....	18
3.1 Розробка удосконаленої моделі якості програмного забезпечення діагностики двигунів.....	19
3.2 Висновки до розділу 3.....	22
4 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ДІАГНОСТИКИ ДВИГУНІВ.....	23
4.1 Ескізний проект програмного забезпечення.....	23
4.1.1 Розробка моделі варіантів використання.....	23
4.1.2 Розробка специфікації варіантів використання.....	23
4.1.3 Розробка сценаріїв варіантів використання.....	25
4.1.4 Розробка прототипу інтерфейсу користувача.....	26
4.2 Технічний проект програмного забезпечення.....	28

4.2.1	Розробка статичної моделі програмного забезпечення.....	28
4.2.2	Розробка специфікації класів програмного забезпечення.....	28
4.2.3	Побудова динамічної моделі програмного забезпечення.....	30
4.3	Робочий проект програмного забезпечення.....	32
4.3.1	Обґрунтування вибору мови програмування та системи програмування.....	32
4.3.2	Реалізація та тестування основних класів програмного забезпечення.....	33
4.3.3	Висновки до розділу 4.....	35
5	РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	37
6	РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	38
6.1	Розрахунок витрат на створення й експлуатацію програми.....	38
6.2	Розрахунок економічної ефективності розробки.....	41
6.3	Висновки до розділу 6.....	41
7	ОХОРОНА ПРАЦІ.....	43
7.1	Основні положення та норми охорони праці.....	43
7.2	Аналіз шкідливих та небезпечних факторів, характерних для офісного приміщення.....	44
7.3	Розрахунок умов праці в офісному приміщенні.....	46
7.4	Розробка заходів щодо забезпечення сприятливих умов праці при роботі з персональним комп'ютером у офісному приміщенні.....	49
7.5	Висновки до розділу 7.....	54
8	ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА.....	55
8.1	Основні положення охорони навколишнього середовища	55
8.2	Забруднення навколишнього середовища підприємством. Заходи щодо зменшення забруднення навколишнього середовища.....	56
8.3	Висновки до розділу 8.....	59
	ВИСНОВКИ	61
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62

Додаток А Технічне завдання.....	65
Додаток Б Текст програми.....	70
Додаток В Опис програми.....	73
Додаток Г Інструкція користувача.....	74
Додаток Д Програма і методика випробувань програмного забезпечення...	75

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

ЕД	емпіричні дані
ПЗ	програмне забезпечення
ЕБК (ЕБУ)	Електронний блок керування
LOC	Lines Of Code

ВСТУП

Актуальність теми. Сучасний кваліфікований фахівець діагност не може обійтись без навичок володіння програмним забезпеченням для діагностики двигунів. Ці програмні продукти є достатньо складними і тому витратними у придбанні і впровадженні. Застосування моделі якості в розробці програмного забезпечення діагностики двигунів дозволяє оптимізувати цикл розробки програмного забезпечення та вигоди від його використання.

Модель якості у використанні програмного забезпечення включає групи таких критеріїв, що достатньо всебічно характеризують ПЗ із позицій адаптованості до галузі застосування і подальшого вдосконалення за побажанням клієнта.:

1. Функціональність – найважливіший критерій. Наявність функцій відрізняє ПЗ від інших, а також робить його корисним.
2. Надійність – відсутність помилок у ПЗ і стійкість до помилок при використанні.
3. Зручність використання. – зрозумілість, зручні процедури, легкість навчання користувачів, привабливість.
4. Ефективність – швидкість отримання результатів, потрібні для роботи ресурси.
5. Адаптованість до різного операційного середовища - здатність пристосовуватися до різних ОС без потреби в оновленні технічних засобів і зміні системного програмного забезпечення.
6. Співвідношення з іншими програмами – здатність співіснувати з іншими програмами в загальному оточенні, підтримка плагінів, що забезпечує розширення функціональних можливостей.

Кількісні значення характеристик передбачено визначати за метриками стандартів на основі вимірів і експертних оцінок. Ефективний процес

оцінювання характеристик якості визначено у стандарті ISO/IEC з оцінювання якості програмних продуктів.

Актуальність проблеми отримання ефективної моделі якості програмного забезпечення діагностики двигунів є важливим завданням. Адже саме якість може стати важливою характеристикою для успіху серед конкурентів на ринку ще на ранньому етапі розробки.

Мета і завдання дослідження. Покращення якісних характеристик програмного забезпечення для діагностики двигунів та їх реалізації.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- Дослідити сучасні моделі якості. з точки зору відповідності до їх застосуванню у програмному забезпеченні діагностики двигунів.
- Створити модель якості для оцінки проектних рішень при розробці програмного забезпечення діагностики
- Розробити проект програмного забезпечення..... з урахуванням оцінок на основі моделі якості.

Об'єктом дослідження є програмне забезпечення для діагностики двигунів.

Предметом дослідження є якість програмного забезпечення для діагностики двигунів

Наукова новизна одержаних результатів полягає в удосконаленні існуючої моделі якості шляхом урахування особливостей програмного забезпечення для діагностики двигунів.

Практичне значення одержаних результатів. Результатом роботи було отримання програмного забезпечення діагностики двигунів покращеного за допомогою удосконаленої моделі якості.

Особистий внесок здобувача. Магістерська робота є самостійно виконаною працею. Усі результати, викладені у роботі, отримані автором особисто.

1 ДІАГНОСТИКА ДВИГУНІВ: СУЧАСНІ ТЕХНОЛОГІЇ

1.1 Програмне забезпечення діагностики двигунів

У сучасні автомобілі вбудовані електронні прилади, які відстежують і контролюють роботу авто. Якщо виникає проблема, не завжди відразу виходить виявити її, тим більше - визначити причину. Тут потрібна професійна автодіагностика з використанням певного спеціального і універсального обладнання. Іноді порушення в роботі агрегатів неможливо виявити через несправні індикаторів і датчиків. Вони відображають показники, які не відповідають реальній картині. Ось чому потрібна систематична діагностика автомобіля.

Перевіряти справність систем, агрегатів і вузлів автомобіля потрібно не рідше, ніж один раз на рік. Також це варто зробити: перед покупкою авто з рук, щоб оцінити його стан; перед ремонтом однієї з систем, щоб виключити поломки в інших вузлах; після перенесеного ДТП.

Така процедура в майбутньому заощадить суму, несумірну з тим, скільки коштує комплексна діагностика і перевірка роботи авто. Оскільки агрегати автомобіля взаємопов'язані електропроводкою, активуються і регулюються електронікою, починати перевірку доцільно з комп'ютерної діагностики. Так виходить швидше знайти пошкодження.

Функціонально комп'ютерна діагностика проводить електронну інспекцію систем автомашини і виводить отримані дані у вигляді графічних показників, а також у вигляді кодів помилок, за допомогою чого можна виправляти поломки або попереджати їх.

Навіть на початковому рівні оволодіння навичками комп'ютерної діагностики, крім іншого, буде доступно:

- Перевірка якості проведеного техобслуговування;

- Більш точне планування автосервісних робіт і економія бюджету;
- Більш точне визначення стану машини при її покупці;
- Самостійне визначення характеру несправностей при сигналі лампи «Check engine».

Існуючі програми OBD діагностики

Функціонал програм може бути дуже різний. Ось деяка частина того, що є в більшості додатків, для загального уявлення:

- Виведення на екран базових параметрів і умов роботи всіх систем автомобіля;
- Зчитування кодів несправності;
- Розшифровка кодів помилок;
- Стирання помилок після усунення їх причини;
- Визначення та коментування статусу помилок;
- Ведення всіляких звітів, журналів техобслуговування.

1.2 Існуючі програмні рішення для діагностики двигунів

На ринку програмного забезпечення для діагностики двигунів існують наступні варіанти програмного забезпечення:

- VAG Tool - Програма призначена для діагностики автомобілей концерну VAG – VW, Audi, Seat, Skoda. Можливості VAG Tool читання кодів несправностей, читання потоків даних, активні тести виконавчих вузлів.

- VDS-PRO – - Програма призначена для діагностики автомобілів групи VAG. Можливості VDS-PRO: перевірка накопичених повідомлень про несправності; діагностика виконавчих пристроїв; накопичених повідомлень про несправності; декодування пристрою управління; читання груп вимірюваних величин; адаптація каналів.

- VAG EEPROM Programmer – Програма спеціально для роботи з автомобілем концерну VAG. Працює через адаптер K-Line. Можливості VAG EEPROM Programmer : корекція пробігу; читання логіна від іммобілайзера; сброс помилок SRS Airbag; читання і сброс кодів помилок.

- ScanXL Professional – Програма для діагностики автомобілів марки Ford, Lincoln, Mercury, GM, Mazda – по протоколу OBD2 для адаптера ELM327. Програма ScanXL Professional не є дилерською програмою, але має в своїй базі достатньо багато заводських параметрів автовиробників. Крім заводських параметрів є велика кількість параметрів, розрахункових параметрів. В тому числі, можна самому створювати такі параметри.

У більшості своїй, програми діагностики двигунів розробляються леше для своєї групи марок виділених по основним авто-концернам. Тому існує потреба у створенні більш загальних програм для користування не спеціалістами. Але більшість нині існуючих програм мають велику різницю в якості одна від одної.

1.3 Висновки до розділу 1

Програмне забезпечення для діагностики двигунів взаємодіє з ЄБУ двигуна.

Програми діагностики двигунів кардинально відрізняються між собою в якості. Тому доцільно удосконалити модель якості для виконання ПЗ діагностики.

2 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ЯКОСТІ ПРОГРАМНОГО ЗАБЕПЕЧЕННЯ

2.1 Характеристика якості програмного забезпечення діагностики двигунів

Стандарт ISO/IEC 9126 регламентує зовнішні і внутрішні характеристики якості. Перші відображають вимоги до функціонування програмного продукту. Для кількісного встановлення критеріїв якості, за якими буде здійснюватися перевірка і підтвердження відповідності ПЗ заданим вимогам, визначаються відповідні зовнішні вимірювані властивості (зовнішні атрибути) ПЗ, метрики (наприклад, час виконання окремих компонентів), діапазони зміни значень і моделі їх оцінки. Метрики використовуються на стадії тестування або функціонування і називаються зовнішніми метриками. Вони являють собою моделі оцінки атрибутів.

Внутрішні характеристики якості і внутрішні атрибути ПЗ використовуються для складання плану досягнення необхідних зовнішніх характеристик якості продукту. Для квантифікації внутрішніх характеристик якості застосовують внутрішні метрики, як інструмент перевірки відповідності проміжних продуктів внутрішнім вимогам до якості, які формулюються на процесах, що передують тестуванню.

Зовнішні і внутрішні характеристики якості відображають властивості самого ПЗ (працюючого або не працюючого), а також погляд замовника і розробника на таке ПЗ. Безпосереднього кінцевого користувача ПЗ цікавить експлуатаційна якість ПЗ – сукупний ефект від досягнення характеристик якості, що вимірюється строком результату, а не властивістю самого ПЗ. Це поняття ширше, ніж будь-яка окрема характеристика (наприклад, зручність використання або надійність).[14]

2.2 Існуючі моделі якості програмного забезпечення.

Моделі мають різну кількість рівнів і повністю або частково збігаються щодо набору характеристик якості. Наприклад, модель якості МакКолла на найвищому рівні має три характеристики: функціональність, модифікованість і портативність, а на нижчих рівнях моделі – 11 характеристик якості і 18 критеріїв (атрибутів) якості. Стандарт ISO 9126 пропонує використовувати для опису внутрішнього та зовнішнього якості ПЗ багаторівневу модель. На верхньому рівні виділено 6 основних характеристик якості ПЗ. Кожна характеристика описується за допомогою кількох вхідних у неї атрибутів. Для кожного атрибута визначається набір метрик, що дозволяють його оцінити. Множина характеристик і атрибутів якості згідно з ISO 9126.[14]

Визначення цих характеристик і атрибутів за стандартом ISO 9126:2001

1. Функціональність – здатність ПЗ в певних умовах вирішувати задачі, потрібні користувачам. Визначає, що саме робить ПЗ, які задачі воно вирішує.

- Функціональна придатність – здатність вирішувати потрібний набір задач.

- Точність – здатність видавати потрібні результати.

- Захищеність – здатність запобігати неавторизованому, тобто без вказівки особи, що намагається його здійснити, і недозволеному доступу до даних і програм.

2. Надійність – здатність ПЗ підтримувати визначену працездатність у заданих умовах.

3. Зручність використання – здатність ПЗ бути зручним у навчанні та використанні, а також привабливим для користувачів.

- Зрозумілість – показник, зворотний до зусиль, які затрачаються користувачами на сприйняття основних понять ПЗ та усвідомлення їх застосовності для розв'язання своїх задач.

- Зручність роботи – показник, зворотний зусиллям, що вживається користувачами для розв'язання своїх задач за допомогою ПЗ.

4. Ефективність – здатність ПЗ при заданих умовах забезпечувати необхідну працездатність стосовно виділюваного для цього ресурсам. Можна визначити її і як відношення одержуваних за допомогою ПЗ результатів до затрачуваних на це ресурсів усіх типів.

5. Зручність супроводу – зручність проведення всіх видів діяльності, пов'язаних із супроводом програм.

6. Портативність – здатність ПЗ зберігати працездатність при перенесенні з одного оточення в інше, включаючи організаційні, апаратні й програмні аспекти оточення.

- Адаптованість – здатність ПЗ пристосовуватися різним оточенням без проведення для цього дій, крім заздалегідь передбачених.

- Зручність установки – здатність ПЗ бути встановленим або розгорнутим у певному оточенні.

- Здатність до співіснування – здатність ПЗ співіснувати з іншими програмами у загальному оточенні, ділячи з ними ресурси.[14]

Загальна модель якості, визначена у стандарті, не враховує особливості задач діагностики двигунів і її доцільно вдосконалити шляхом врахування специфіки цього виду програмного забезпечення.

2.3 Шляхи удосконалення моделі якості програмного забезпечення діагностики двигунів.

Так як моделі якості програмного забезпечення для діагностики не існує, необхідно таку модель створити.

Як основу для нової моделі якості візьмемо стандарт ISO/IEC 9126. Стандарт ISO/IEC 25010 пропонує використовувати для опису внутрішнього

та зовнішнього якості ПЗ багаторівневу модель. На верхньому рівні виділено 6 основних характеристик якості ПЗ. Кожна характеристика описується за допомогою кількох вхідних у неї атрибутів. Для кожного атрибута визначається набір метрик, що дозволяють його оцінити. З них треба обрати найбільш вагомі для програмного забезпечення діагностики.

Значна частина метрик характеристик, що використані у моделі якості визначається за методом експертних оцінок. Оцінювання ПЗ проводиться двома групами експертів: статичні експерти і динамічні експерти. До першої групи відносяться авторитетні фахівці, наприклад, спеціаліст галузі, в якій буде застосовано ПЗ, і юзабіліті фахівець для оцінювання зрозумілості застосування. До другої групи відносяться кінцеві користувачі програмного забезпечення, зауваження і побажання яких безпосередньо впливають на якість і зручність використання програмного забезпечення. Оцінки таких експертів можуть дещо різнитись. Відповідно, оцінкам кожної групи експертів може бути призначена певна вага.

2.4 Висновки до розділу 2

Розроблено шлях створення модель якості програмного забезпечення для багатокритеріального оцінювання програмного забезпечення діагностики двигунів, яка використовує експертні оцінки і може бути налаштована відповідно до контексту використання і преференцій користувачів.

3 УДОСКОНАЛЕННЯ МОДЕЛІ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДІАГНОСТИКИ ДВИГУНІВ

3.1 Розробка удосконаленої моделі якості програмного забезпечення діагностики двигунів

Для покращення якості виконання проектів програмного забезпечення діагностики двигунів необхідно розробити модель якості, яка буде відповідати основним потребам основних користувачів ПЗ.

Модель розроблюється на основі стандарту ISO/IEC 9126, що визначає характеристики якості програмного забезпечення. Структура моделі і номенклатура характеристик сформована на основі зовнішніх характеристик якості і характеристик якості при використанні з урахуванням рекомендацій.

Для успішного контролю якості необхідні формалізовані, кількісні оцінки показників якості. Такими оцінками виступають метрики ПЗ. Вони знаходяться на найнижчому – фундаментальному рівні моделі якості ПЗ та є основоположними. Саме тому починати оцінювання якості потрібно з та базуючись на метриках.

Для створення моделі якості програм діагностики двигунів, виходячи з аналізу існуючих програм діагностики, як ключові було обрано наступні характеристики:

1. Результативність - це здатність програмного продукту, що дозволяє користувачам досягати заданий мети з точністю і повнотою в заданому контексті використання.

2. Продуктивність - це здатність програмного продукту, що дозволяє користувачам витрачати кількість ресурсів, відповідне результативності, що досягається в заданому контексті використання. Ресурси можуть включати час виконання завдання, зусилля користувача, матеріали, вартість використання.

3. Функціональність – здатність ПЗ в певних умовах вирішувати задачі, потрібні користувачам. Визначає, що саме робить ПЗ, які задачі воно вирішує.

- придатність;
- коректність (правильність, точність);
- здатність до взаємодії;
- захищеність;
- відповідність функціональності.

4. Зручність використання – здатність ПЗ бути зручним у навчанні та використанні, а також привабливим для користувачів.

- зрозумілість;
- здатність до навчання;
- простота використання;
- привабливість;
- відповідність практичності.

Кількісні значення характеристик передбачено визначати за метриками стандартів на основі вимірів і експертних оцінок. Ефективний процес оцінювання характеристик якості визначено у стандарті ISO/IEC з оцінювання якості програмних продуктів.

Внутрішні метрики - це метрики, що вимірюють власні властивості програмного засобу. Вони вимірюються в процесі розробки програмного засобу на основі специфікації вимог, результатів проектування, вихідного коду або іншої документації програмного засобу. Внутрішні метрики дають можливість оцінити якість проміжних програмних продуктів розробки, передбачаючи якість кінцевого програмного засобу.

Внутрішні метрики функціональності призначені для передбачення того, чи задовольняє розробляється програмний продукт вимогам до функціональності і очікуваним потребам користувача.

1. Повнота функціональної реалізації, оцінює придатність ПЗ.
2. Відповідність інтерфейсів, оцінює здатність до взаємодії.

Внутрішні метрики практичності використовуються під час розробки програмного продукту для передбачення ступеня, в якій програмний продукт може бути зрозумілий, вивчений, керований, привабливий і відповідає домовленостям і інструкцій з практичності

1. Здатність до демонстрації, оцінює зрозумілість.
2. Повнота документації користувача, оцінює здатність до навчання.
3. Настроюваність виду інтерфейсу користувача, оцінює привабливість.

Зовнішні метрики - це метрики, призначений для вимірювання якості програмного продукту шляхом вимірювання поведінки системи, частиною якої є даний продукт. Зовнішні метрики можуть бути використані в процесі експлуатації і на стадіях тестування або випробувань в процесах розробки і супроводу програмного засобу, коли вже створені виконані коди програмного продукту.

Зовнішні метрики функціональності повинні вимірювати властивості (атрибути) функціональної поведінки системи, що містить програмний засіб.

1. Повнота функціональної реалізації, оцінює придатність ПЗ.
2. Здатність до обміну даними (заснована на успішних спробах користувача), оцінює здатність до взаємодії.

Зовнішні метрики практичності показують, якою мірою програмний засіб може бути зрозуміле, вивчено, керовано, привабливо і відповідає домовленостям і інструкцій з практичності.

1. Повнота опису, оцінює зрозумілість.
2. Ефективність документації користувача, оцінює здатність до навчання.
3. Змінність виду інтерфейсу, оцінює привабливість.

Метрики якості програмних засобів у використанні - метрики, що вимірюють відповідність продукту потребам, заданих користувачем, досягнення заданих цілей з результативністю, безпекою, продуктивністю і задоволенням, заданим в контекстах використання. Метрики

використовуються тільки в процесі експлуатації ПЗ в реальному середовищі оточення. І засновані на вимірі поведінки типових користувачів і системи, що містить дане програмний засіб.

Метрики результативності оцінюють, чи досягають завдання, що виконуються користувачем, заданих цілей з точністю і повнотою в заданому контексті використання.

- Завершені задачі

Метрики продуктивності оцінюють ресурси, які витрачають користувачі відповідно до досягнутої результативністю в заданому контексті використання.

- Коефіцієнт продуктивності

3.2 Висновки до розділу 3

Так як програми діагностики двигунів це спеціалізовані програми то вибрані характеристики будуть найкращим чином впливати на якість ПЗ. А обрані метрики допоможуть у його оцінюванні підчас випробувань та тестування.

4 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ДІАГНОСТИКИ ДВИГУНІВ

4.1 Ескізний проект програмного забезпечення

4.1.1 Розробка моделі варіантів використання

Модель варіантів використання представлена на рисунку 4.1. На цій діаграмі варіантів використання виділено 1 актора та відповідні прецеденти:

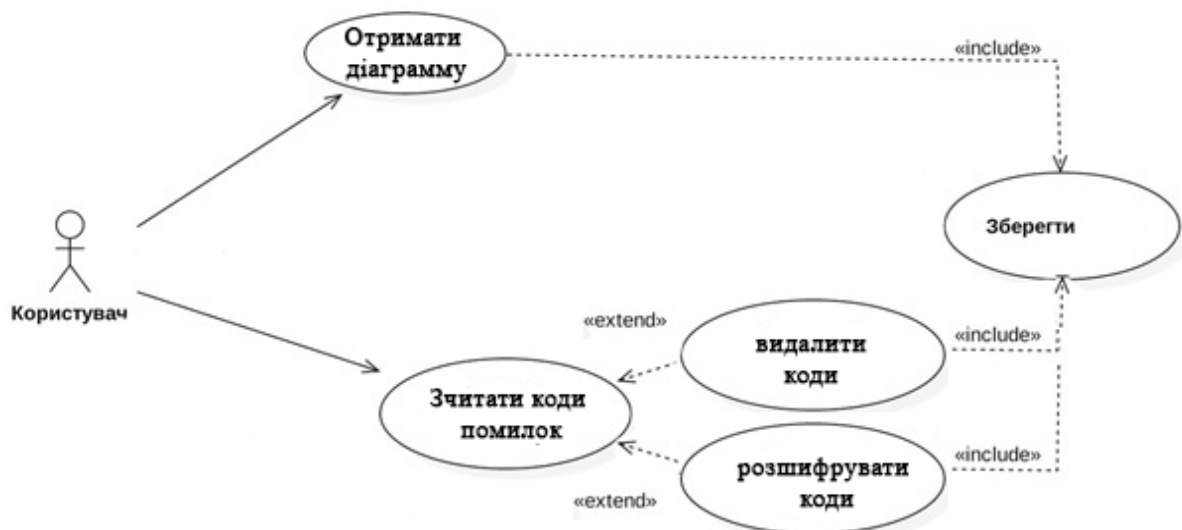


Рисунок 4.1 – Діаграма варіантів використання програмного забезпечення

4.1.2 Розробка специфікації варіантів використання

Потік подій варіанта використання має такі складові:

- короткий опис;
- передумови;
- основний потік подій;
- альтернативний потік подій;
- постумови.

Таблиця 4.1 – Опис варіантів використання системи

Опис	Складова
1	2
Підключення до ЕБУ	
Короткий опис	Дана функція дозволяє завантажити параметри з ЕБУ.
Передумови	Програма запущена та виведена початкова форма.
Основний потік подій	<ol style="list-style-type: none"> 1. Програма виводить форму для підключення. 2. Користувач вибирає модель та підтверджує введення; 3. Програма перевіряє ввімкнення запалювання; 4. Якщо запалення ввімкнене - зчитуються дані.
Альтернативний потік подій	<ol style="list-style-type: none"> 5. Програма виводить форму для підключення. 6. Користувач вибирає модель та підтверджує введення; 7. Програма перевіряє ввімкнення запалювання; 8. Якщо запалення вимкнене – виводиться повідомлення про помилку.
Постумови	Зчитуються дані вихідних вибірок із файлу.
Занесення даних	
Короткий опис	Дана функція дозволяє змінити допустимі поля з даними
Передумови	Програма запущена, поле для вводу доступне для редагування
Основний потік подій	<ol style="list-style-type: none"> 1. Користувач вводить дані 2. Програма обробляє нові дані 3. Програма перевіряє коректність введених даних; 4. Програма приймає нові дані
Альтернативний потік подій	<ol style="list-style-type: none"> 1. Користувач вводить дані 2. Програма обробляє нові дані. 3. Програма перевіряє коректність введених даних; 4. Повідомлення про те, що формат не вірний; 5. Надання можливості повторного введення
Постумови	Оновлення даних
Редагування даних	
Короткий опис	Дана функція розраховує параметри для вихідних показників
Передумови	Користувач підключився до ЕБУ
Основний потік подій	<ol style="list-style-type: none"> 1. Програма розраховує параметри; 2. Програма виводить розраховані параметри;
Альтернативний потік подій	Відсутній.
Постумови	Ввод розрахованої інформації, запис інформації у ЕБУ
Перевірка помилок двигуна.	
Короткий опис	Призначений для перевірки наявності помилок в роботі двигуна
Передумови	Користувач підключився до ЕБУ
Основний потік подій	<ol style="list-style-type: none"> 1. Користувач натиснув кнопку «Помилки» 2. Програма виводить повідомлення про наявність помилок.
Постумови	Вивід помилок.

4.1.3 Розробка сценаріїв варіантів використання

Поведінка об'єкта відбивається в його станах, щоб зобразити це, використовується два види діаграм: Statechart diagram (діаграма станів) і Activity diagram (діаграма активності) [34,35].

На рисунках 4.2-4.3 зображено декілька сценарій варіантів використання для прецеденту «Перевірка вихідних показників»:

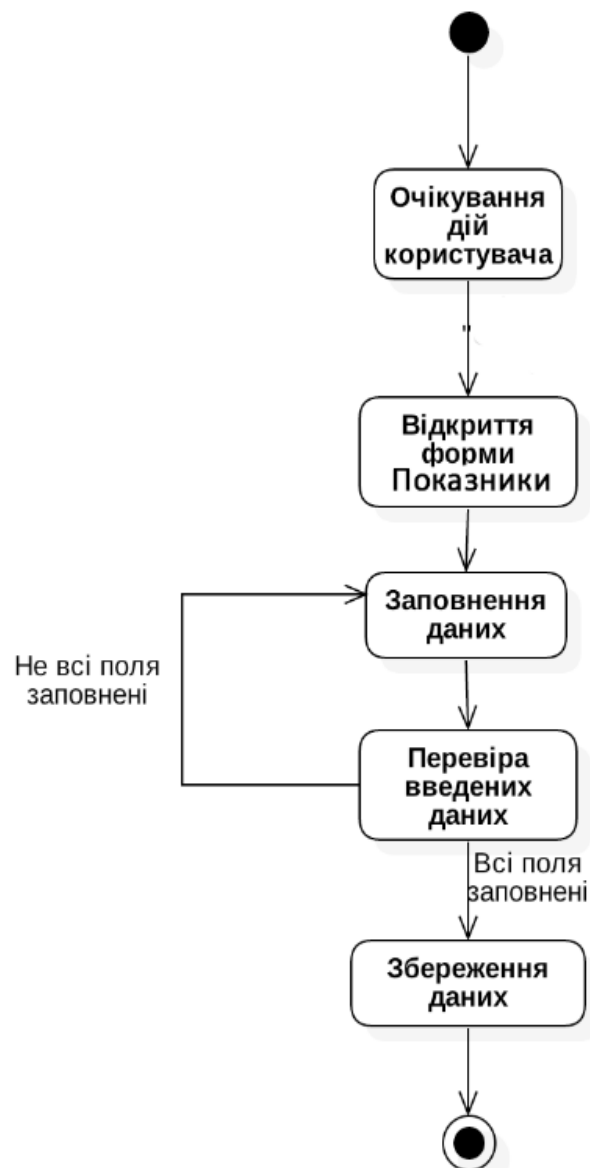


Рисунок 4.2 – Діаграма діяльності для прецеденту «Перевірка вихідних показників»

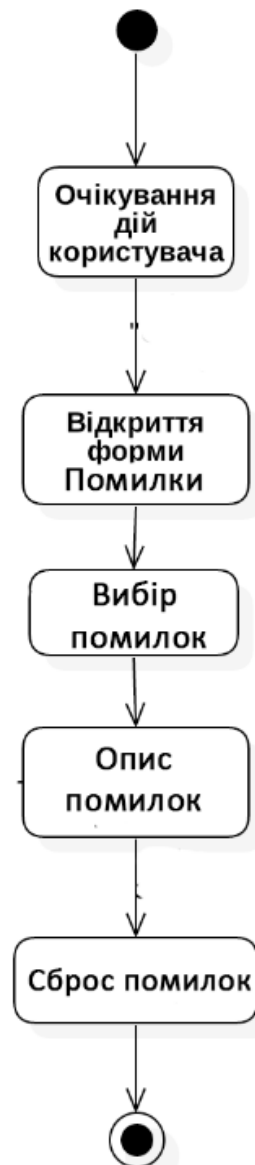


Рисунок 4.3 – Діаграма діяльності для прецеденту «Розшифрування помилок»

4.1.4 Розробка прототипу інтерфейсу користувача

Інтерфейс користувача — сукупність засобів для обробки та відображення інформації, максимально пристосованих для зручності користувача; у графічних системах інтерфейс користувача реалізовується багатовіконним режимом, змінами кольору, розміру, видимості (прозорість, напівпрозорість, невидимість) вікон, їхнім розташуванням, сортуванням

4.2 Технічний проект програмного забезпечення

4.2.1 Розробка статичної моделі програмного забезпечення

Діаграма класів – статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів може містити пакети та вкладені пакети [37]. Діаграма класів програмного забезпечення для діагностики двигунів представлена на рисунку 4.9.

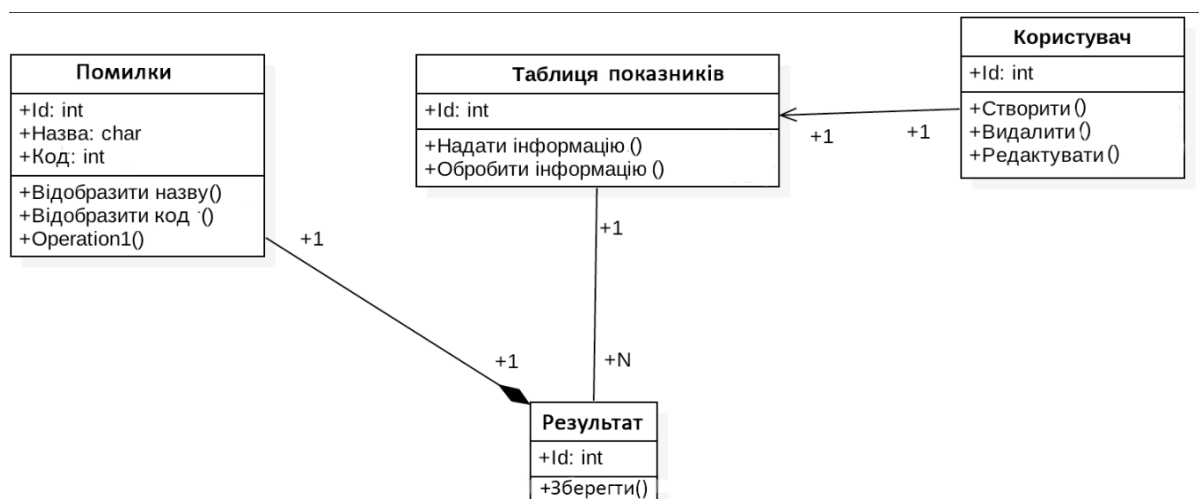


Рисунок 4.6 – Діаграма класів програмного забезпечення для діагностики двигунів

4.2.2 Розробка специфікації класів програмного забезпечення

Специфікація класу для об'єкта – це визначення його властивостей (які характеризують стан об'єкта) і методів (які є способом реалізації його поведінки). Специфікація класів програми представлена в таблиці 4.2.

Таблиця 4.2 – Специфікація класів

Опис	Складова
1	2
Користувач	
Відповідальність	Даний клас відповідає за зчитування показників ЄБУ.
Атрибут	Кількість значень
Операції зі специфікаціями нетривіальних операцій	Створити(): Створити файл Видалити(): видалити параметр Редагувати(): редагувати параметр
Таблиця показників	
Відповідальність	Даний клас відповідає за інформацію про параметри двигуна.
Атрибут	Кількість значень
Операції зі специфікаціями нетривіальних операцій	Надати інформацію(): Ручне регулювання. Обробити інформацію(): Авто корекція
Помилки	
Відповідальність	Даний клас відповідає за інформацію про коди помилок.
Атрибут	Кількість значень ,
Операції зі специфікаціями нетривіальних операцій	Відобразити назву(): Відобразити назву (причину) помилки Відобразити код():Відобразити код.

Продовження таблиці 4.2

1	2
Результат	
Відповідальність	Даний клас відповідає за інформацію про показники і помилки записані в ЄБУ.
Атрибут	Кількість значень ,
Операції зі специфікаціями нетривіальних операцій	Зберегти (): зберегти результати.

4.2.3 Побудова динамічної моделі програмного забезпечення

Динамічна модель програми може бути представлена:

- діаграмами діяльності;
- діаграмами послідовності;
- діаграмами взаємодії;
- діаграмам станів.

Динамічна модель програми представлена у вигляді діаграми послідовності. У магістерській роботі представлені діаграми послідовності деяких варіантів використання.

Діаграма послідовності (sequence diagram) – діаграма, на якій показані взаємодії об'єктів, упорядковані за часом їхнього прояву [38].

Нижче наведено декілька діаграм послідовності для розроблюваного ПЗ на рисунках 4.7 – 4.8.

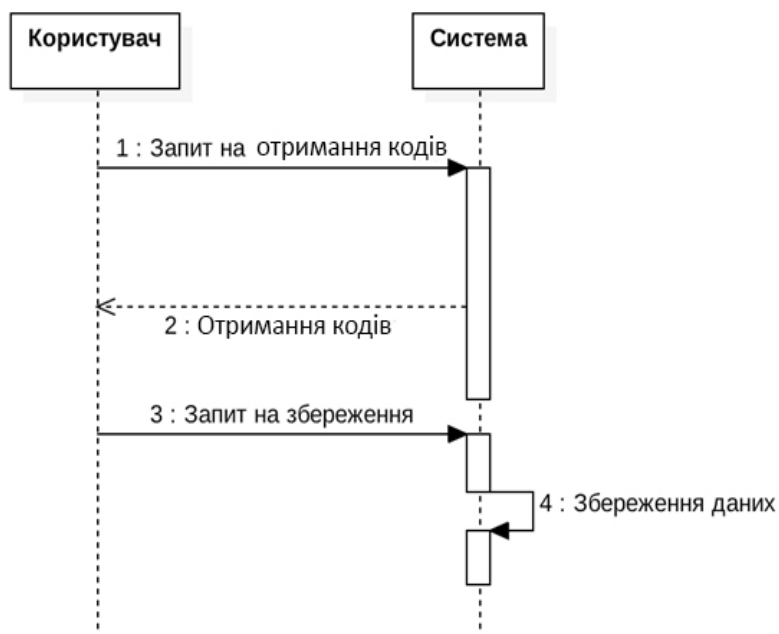


Рисунок 4.7 – Діаграма послідовності процесу «Отримання кодів помилок»

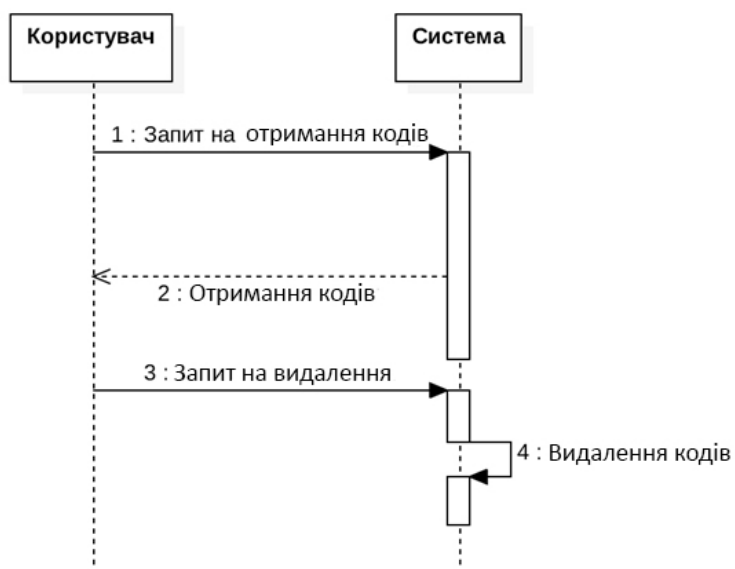


Рисунок 4.8 – Діаграма послідовності процесу “ Видалення кодів помилок ”

4.3 Робочий проект програмного забезпечення

4.3.1 Обґрунтування вибору мови програмування та системи програмування

Для розробки ПЗ було обрано мову програмування Java та середовище розробки NetBeans.

Java – об'єктно-орієнтована мова програмування. Програма на Java транслюється в байт-код, що виконується віртуальною машиною Java (JVM) програмою, яка обробляє байт-код и передає інструкції обладнанню як інтерпретатор.

Переваги мови Java:

1. Охоплює широке коло задач. Завдяки простоті та надійності Java використовують у різних сферах життя: для розробки ПЗ в державній сфері, в науці, освіті, сфері охорони здоров'я, в приватному секторі при створенні програм для трейдингу, серверні додатки для банкінгу та для багатьох інших корпоративних і ентерпрайз цілей.
2. Масштабованість. На мові Java розробляють програми для самих різних платформ і систем.
3. Базується на принципах об'єктно-орієнтованого програмування.
4. Велика кількість бібліотек з відкритим кодом.

Мова Java є однією з наймолодших в сімействі мов програмування і була розроблена з розрахунку на те, щоб професійний програміст міг легко її опанувати та ефективно використовувати. За основу Java взятий синтаксис C++ - безсумнівно однієї з найбільш популярних мов програмування сучасності. Проте, Java - це цілком самостійна мова програмування, і при її створенні не йшлося про будь-яку сумісність з C++. Тому деякі механізми реалізовані в Java інакше, а деякі зовсім відсутні. Ідеологічно ж Java побудована дещо інакше ніж C++. Розробники Java ґрунтувалися на досвіді розробки програм на C++ і прагнули позбутися можливостей, які

zareкомендували себе непевними. Так, в Java відсутня перегрузка операторів а також автоматичне приведення несумісних типів - конструкції, які при неуважному використанні є джерелом важких для виявлення помилок. Взагалі, інтерфейси Java більш прості та прозорі для розуміння. Написати на Java програму з графічним інтерфейсом значно легше. Звичайно, простота інтерфейсів компенсується меншою гнучкістю, бібліотека Java не така багата, як стандартні бібліотеки C/C++. Java задуманий для використання на різних платформах і тому реалізує в собі найбільш стандартні можливості задля легшої адаптації під конкретне середовище. [39].

Найбільш зручною для створення програми мовою Java є NetBeans IDE 8.0.2.

4.3.2 Реалізація та тестування основних класів програмного забезпечення

Реалізація програмного забезпечення була виконана на високорівневій мові програмування Java.

Для тестування програмного забезпечення є такі основні групи методів:

- методи тестування по формальним специфікаціям (методи чорного ящика);
- метод структурного тестування (методи білого ящика).

До групи методів чорного ящика відносяться наступні методи: розбиття на класи еквівалентності та граничні умови та інші. До методів білого ящика відносяться наступні методи: покриття операторів, покриття переходів та інші.

Так як методи структурного тестування вимагають більших затрат ресурсів, то в дипломній роботі виконувалось тестування по специфікаціям, а саме розбиття на класи еквівалентності.

Програму розглядали як чорний ящик. Випробування зводилися до послідовного вводу тестових наборів даних й аналізу отриманих результатів.

В таблиці 4.3 представлені класи еквівалентності та граничні значення вхідних даних:

- тиск;
- температура;
- оберти;
- положення дроселя;

Таблиця 4.3 – Класи еквівалентності вхідних даних для класу таблиця показників

Вхідні умови	Правильні класи еквівалентності	Неправильні класи еквівалентності
тиск	Число	Усі крім числового типу (символьні, логічні, графічні зображення).
температура	Число	Усі крім числового типу (символьні, логічні, графічні зображення).
оберти	Число	Усі крім числового типу (символьні, логічні, графічні зображення).
положення дроселя	Число	Усі крім числового типу (символьні, логічні, графічні зображення).

Почнемо з тестування для правильних класів еквівалентності класу «Таблиця показників», наведених в табл. 4.4.

Таблиця 4.4 – Тести з правильних класів еквівалентності для класу «Таблиця показників»

№	Вхідні умови	Правильні класи еквівалентності	Вихідні дані
1	тиск	В поле «Ню» введено число «25».	Результат вірний.
2	температура	В поле «Гамма» введено число «76».	Результат вірний.
3	оберти	В поле «Гамма» введено число «350».	Результат вірний.
4	положення дроселя	В поле «Гамма» введено число «16.3».	Результат вірний.

Таблиця 4.5 – Тести з неправильних класів еквівалентності класу
«Таблиця показників»

№	Вхідні дані	Вихідні дані
1	В поле «тиск» введемо літеру українського алфавіту.	Програма виводить повідомлення про помилку. Результат підтверджено.
2	В поле «температура» введемо літеру англійського алфавіту.	Програма виводить повідомлення про помилку. Програма далі працює правильно. Результат підтверджено.
3	Поле «оберти» залишимо пустим.	Після введення цифр виводиться повідомлення про помилку. Програма далі працює правильно. Результат підтверджено
4	В поле «положення дроселя» ведемо графічний елемент – смайлик (скопюємо з інтернету).	Після введення смайлика виводиться повідомлення про помилку. Програма далі працює правильно. Результат підтверджено.

Отже, за результатами тестування неправильних класів еквівалентності класу «Таблиця показників», додаток працює правильно, не допускаючи до введення будь-яких інших елементів, окрім стандартних символів вводу. Тестування інших класів виконується аналогічно.

4.3.3 Висновки до розділу 4

UML – це мова, призначена для візуалізації, специфікації, конструювання й документування програмних систем. Крім того її можна використати для моделювання різного роду програм. Для графічного відображення майбутнього ПЗ для діагностики двигунів, побудовано UML діаграму, а саме - діаграму варіантів використання.

Потік подій варіанта використання має такі складові: короткий опис; передумови; основний потік подій; альтернативний потік подій; постумови.

Було розроблено специфікації варіантів використання та специфікації класів програми.

Програмне забезпечення для діагностики двигунів, є програмним забезпеченням такого типу структури, яке не потребує функціональності бази даних. Усі входні дані програма буде отримувати за допомогою підключення до ЕБУ та зчитування з нього.

Після розрахунків усі отримані параметри будуть записуватись у вихідний файл у форматі CSV.

Поведінка об'єкта відбивається в його станах, щоб зобразити це, використовується два види діаграм: Statechart diagram (діаграма станів) і Activity diagram (діаграма активності)

Інтерфейс користувача — сукупність засобів для обробки та відображення інформації, максимально пристосованих для зручності користувача. На рисунках 4.4 – 4.5 було представлено прототип користувацького інтерфейсу програмного забезпечення.

Вказано атрибути та їх типи, а також методи.

Динамічна модель ПЗ представлена діаграмами послідовності на рисунках 4.7 – 4.8.

Для розробки програмного забезпечення було обрано мову програмування Java та середовище розробки NetBeans.

5 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«Програмне забезпечення діагностики двигунів» засноване на удосконаленій моделі якості ПЗ..

Дане програмне забезпечення може використовуватись як великими компаніями з ремонту авто транспорту, так і просто окремими фахівцями. Воно потребує деякого рівня навичок у сфері діагностування.

Усі нижчезазначені функції були реалізовані в повному обсязі у розробленій програмі та випробувані під час етапу тестування:

- можливість завантажити файл з вихідними даними у форматах Excel (.xml, .xls, .xlsx);
- можливість змінити файл з вихідними даними;
- зчитування даних з файлу;
- розрахунок параметрів для регулювання;;

Інтерфейс був розроблений максимально простим для зручності користувача, кольорова гамма приємна та спокійна для очей. Шрифт достатньо великий.

Також було реалізовано функцію завантаження файлу з розрахованими параметрами у форматі CSV.

Усі вимоги наведені у технічному завданні (див. Додаток А) були виконані. В якості технічних ресурсів був використаний ПК з комплектацією наведеною у технічному завданні.

Розробка програми, як вказано у техніко-економічному розділі, потребує 1,5 місяці роботи програміста та витрати у розмірі 32101,64 грн. на створення й експлуатація програми за перший рік. Строк окупності ПЗ для оцінювання часу на виконання веб-додатків, реалізованих мовою Liquid, складатиме приблизно півтора місяці.

6 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Швидка та легка взаємодія людини з інформаційною системою – це одна з найважливіших задач, яку необхідно виконати під час розробки програмного забезпечення.

Дана магістерська робота присвячена розробці програмного забезпечення для вдосконалення математичної моделі для діагностики двигунів.

Обґрунтувати доцільність розробки з економічної точки зору можливо, розрахувавши собівартість програмного забезпечення та прибуток, що буде отриманий в результаті продажу. Основним показником, що визначає економічну доцільність витрат на створення продукту – є річний економічний ефект. Характерним показником економічної ефективності створення і функціонування програмного продукту, що розраховується на стадії створення і впровадження – є ефективність витрат, яку характеризує строк окупності програмного забезпечення.

У даному підрозділі розраховується собівартість розробки програмного забезпечення. Також виконується оцінка ринкової ціни та об'ємів продажу копій програми. На основі цих даних розраховується річний економічний ефект та строк окупності розробки.

6.1 Розрахунок витрат на створення й експлуатацію програми

Витрати на розробку програмного забезпечення складаються з витрат на заробітну плату розробника, на амортизацію та експлуатацію ЕОМ, на якій виконується розробка, на засоби розробки та витрати на матеріали і комплектуючі.

Розробка програмного забезпечення виконується спеціалістом, місячний оклад якого складає 15000 грн. Додаткова заробітна плата складає 10% від основної. Виходячи з цього, основна і додаткова заробітна плата розроблювача системи 16500 грн./міс, а вартість сучасного ноутбуку складає 17000 грн. (середня вартість машини на базі Core i5). Вартість кіловат-години електроенергії дорівнює 1,68 грн. Витрати на допоміжні матеріали наведені в табл. 6.1

Таблиця 6.1 – Витрати на допоміжні матеріали

Елементи витрат	Сума
Папір	90
Заправлення картриджа до принтера	130
Непередбачені витрати	500
Разом матеріали і комплектуючі вироби.	415
Папір	90

Вартість розробки системи розраховується за формулою:

$$C_{\text{пр}} = (Z_{\text{зн}} + Z_{\text{сз}} + Z_{\text{зг}} + Z_{\text{е}}) * T + Z_{\text{м}} \quad (6.1)$$

де T - тривалість розробки, міс.

$Z_{\text{зн}}$ - основна і додаткова заробітна плата обслуговуючого персоналу, грн.;

$Z_{\text{сз}}$ - відрахування на соціальні заходи (38% від основної і додаткової заробітної плати), грн.;

$Z_{\text{зг}}$ - загальногосподарські витрати (10% від заробітної плати), грн.;

$Z_{\text{е}}$ - витрати на електроенергію;

$Z_{\text{м}}$ - витрати на основні і допоміжні матеріали.

Розрахуємо $Z_{\text{е}}$ при споживанні потужності 0,3 Вт, тривалості роботи на місяць рівної $21 * 8 = 168$ годин і вартості кіловат-години електроенергії 1,68 грн.,

$$Z_{\text{е}} = 168 * 1,68 * 0,3 = 84,67 \text{ грн.}$$

Представимо всі поточні витрати на розробку програмного забезпечення в таблиці 6.2

Таблиця 6.2 - Витрати на розробку програмного забезпечення

Найменування витрат	Одиниця виміру	Кількість
Тривалість розробки	місяць	1
Основна і додаткова заробітні плати	грн.	16500
Відрахування на соціальні заходи	грн.	1743
Загальногосподарські витрати	грн.	1500
Витрати на допоміжні матеріали	грн.	415
Витрати на електроенергію	грн.	84,67

Відповідно до формули (1) вартість розробки ПЗ складас:

$$C_{\text{пр}} = 16500 + 1743 + 1500 + 415 = 20158 \text{ грн.}$$

Амортизаційні відрахування на устаткування складають 60% балансової вартості в рік:

$$A_{\text{об}} = 17000 * 0,6 = 10200 \text{ грн.}$$

У масштабах підприємства річні витрати на основні і допоміжні матеріали визначаються в розмірі 5% вартості основного устаткування:

$$B_{\text{м}} = 9000 * 0,05 = 850 \text{ грн.}$$

Річний обсяг робіт ПЕОМ у годинах визначається в такий спосіб:

$$\Phi_{\text{м}} = 253,3 * T_3$$

де T_3 - це середнє місячне завантаження устаткування (близько 7 годин), 253,3 - середня кількість робочих днів у році:

$$\Phi_{\text{м}} = 253,3 * 7 = 1773,1 \text{ год.}$$

Витрати на електроенергію Z_e при 1773,1 годинах роботи устаткування в рік складуть:

$$Z_e = 1773,1 * 0,3 * 1,68 = 893,64 \text{ грн.}$$

Експлуатаційні витрати для ПЕОМ за рік складуть:

$$Z_{\text{зр}} = 10200 + 850 + 893,64 = 11943,64 \text{ грн.}$$

В перший рік витрати на створення й експлуатацію програмного забезпечення складуть:

$$З_{се} = 20158 + 11943,64 = 32101,64 \text{ грн.}$$

6.2 Розрахунок економічної ефективності розробки

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження програмного забезпечення вивільняє одного працівника (за експертною оцінкою фахівців підприємства).

Зарплата одного працівника в рік складає:

$$15000 * 12 * 1 = 180000 \text{ грн.}$$

Річний економічний ефект розраховується по формулі:

$$= \quad , \quad (6.2)$$

де ΔC_n – вивільнені кошти після впровадження системи (180000 грн.) мінус експлуатаційні витрати (11943,64 грн.);

E_n – коефіцієнт ефективності (дорівнює коефіцієнту амортизації (0,6));

k – одноразові витрати на впровадження продукту (20158).

$$\mathcal{E}_{\text{рік}} = 180000 - 11943,64 - 20158 * 0,6 = 155961,56 \text{ грн.}$$

Строк окупності системи розраховується по формулі:

$$T = \frac{k}{\Delta C} = \frac{20158}{168056,36} \approx 0,12 \text{ року}$$

Отже строк окупності програмного забезпечення складає приблизно 1.5 місяця.

6.3 Висновки до розділу 6

Розрахунок вартості програмного продукту та процес формування витрат є дуже важливим моментом для розробника. Саме тому необхідно провести розрахунок витрат на створення й експлуатацію майбутнього ПЗ.

Витрати на розробку ПЗ охоплюють такі витрати:

- заробітна плата (ЗП) розробника;

- амортизація та експлуатація ЕОМ, на якій виконується розробка;
- засоби розробки, матеріали, комплектуючі і т.д.

Основним показником економічної ефективності функціонування ПЗ є підвищення ефективності керування інформацією у вигляді зниження витрат на керування при одночасному збільшенні швидкості і якості одержання потрібного результату.

Створення й експлуатація програми за перший рік нараховуватиме 32101,64 грн. Строк окупності програмного забезпечення для діагностики двигунів, складатиме приблизно 1.5 місяця.

7 ОХОРОНА ПРАЦІ

7.1 Основні положення та норми охорони праці

Законодавство України про охорону праці являє собою систему взаємозв'язаних нормативно-правових актів, що регулюють відносини у галузі реалізації державної політики щодо правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці. Воно складається з Закону України «Про охорону праці», Кодексу законів про працю України, Закону України «Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності» та прийнятих відповідно до них нормативно-правових актів.

Базується законодавство України про охорону праці на конституційному праві всіх громадян України на належні, безпечні і здорові умови праці, гарантовані статтею 43 Конституції України.

Охорона праці - система забезпечення безпеки життя і здоров'я працівників у процесі трудової діяльності, що включає правові, соціально-економічні, санітарно-гігієнічні, психофізичні, лікувально-профілактичні, реабілітаційні та інші заходи.

Функціями охорони праці є дослідження санітарії та гігієни праці, проведення заходів щодо зниження впливу шкідливих факторів на організм працівників у процесі праці. Основним методом охорони праці є використання техніки безпеки.

Техніка безпеки - система організаційних і технічних заходів і засобів, що запобігають вплив на працюючих небезпечних виробничих факторів. [41]

7.2 Аналіз шкідливих та небезпечних факторів, характерних для офісного приміщення

Шкідливим фактором називається такий фактор, вплив якого на робітника в певних умовах призводить до захворювання або зниження працездатності.

Небезпечний фактор – фактор, дія якого за певних умов може призвести до травм або іншого раптового погіршення здоров'я працівника.

Відповідно до ГОСТ 12 0.003-74 небезпечні та шкідливі фактори за природою дії поділяються на такі групи:

- фізичні;
- хімічні;
- біологічні;
- психофізіологічні.

Фізичні виробничі фактори поділяються на:

- рухомі машини та механізми;
- рухомі частини виробничого обладнання;
- шум;
- вібрацію (загальну і локальну);
- інфразвук;
- ультразвук;
- електромагнітні поля радіочастотного діапазону;
- електричні поля промислової частоти;
- електростатичні поля;
- лазерне випромінювання;
- іонізуюче випромінювання;
- освітленість;
- ультрафіолетове випромінювання;
- мікроклімат у виробничому приміщенні (температура повітря, швидкість руху повітря, відносна вологість повітря, інтенсивність

- інфрачервоного (теплого) випромінювання);
- аероіонізація повітря;
- атмосферний тиск (підвищений, знижений).

Хімічні виробничі фактори поділяються за характером впливу на організм людини на:

- токсичні;
- дратівливі;
- канцерогенні;
- мутагенні і т.д.

Хімічні виробничі фактори за способом проникнення в організм людини поділяються на проникаючі через:

- органи дихання;
- шлунково-кишковий тракт;
- шкірні покриви і слизові оболонки.

Біологічні виробничі фактори включають біологічні об'єкти:

- патогенні мікроорганізми (бактерії, віруси, гриби найпростіші і т.д.)

та продукти їх життєдіяльності;

- мікроорганізми (рослини, тварини);
- природні компоненти організму.

Психофізіологічні виробничі фактори визначаються показниками важкості та напруженості трудового процесу.

Основними показниками важкості трудового процесу є:

- величина фізичної динамічного навантаження;
- разова величина вантажу, що піднімається вручну (маса що піднімається і переміщуваного вантажу вручну);

- статичне навантаження;
- робоча поза і переміщення в просторі (робоча поза; нахили корпусу;
- переміщення в просторі, обумовлені технологічним процесом);
- темп роботи (стереотипні робочі руху);

Основними показниками напруженості трудового процесу є:

- інтелектуальні навантаження;
- напруженість уваги (сенсорні навантаження);
- напруженість функцій аналізаторів (сенсорні навантаження);
- монотонність (монотонність навантажень);
- емоційне напруження (емоційні навантаження);
- естетичний дискомфорт;
- фізіологічний дискомфорт;
- змінність (режим роботи).

7.3 Розрахунок умов праці в офісному приміщенні

До сучасного виробничого освітлення пред'являються високі вимоги як гігієнічного, так і техніко-економічного характеру. Правильно спроектоване і виконане освітлення забезпечує високий рівень працездатності, надає позитивний психологічний вплив на працюючих, сприяє підвищенню продуктивності праці.

До систем освітлення висувають наступні вимоги:

- відповідність рівня освітленості робочих місць характеру виконуваної зорової роботи;
- досить рівномірний розподіл яскравості на робочих поверхнях і в навколишньому просторі;
- сталість освітленості в часі;
- оптимальна спрямованість випромінюваного освітлювальними приладами світлового потоку;
- довговічність, економічність, електро- та пожежобезпечність, естетичність, зручність і простота експлуатації.

У тих випадках, коли одного природного освітлення в приміщенні недостатньо, створюють поєднане освітлення. При цьому додаткове штучне освітлення застосовують не тільки в темний, але й у світлий час доби.

По конструктивному виконанню штучне освітлення може бути загальними і місцевим. При загальному освітленні всі робочі місця одержують освітлення від загальної освітлювальної установки. Комбіноване освітлення поряд із загальним включає місцеве освітлення, яке зосереджує світловий потік безпосередньо на робочих місцях. Застосування тільки місцевого освітлення неприпустимо, тому що виникає необхідність частоті адаптації зору, створюються глибокі і різкі тіні та інші несприятливі чинники. Для штучного освітлення приміщень використовують люмінесцентні лампи, у яких висока світлова віддача і тривалий термін служби. [42]

Розрахунок штучного освітлення будемо виконувати методом коефіцієнта використання світлового потоку.

Розрахуємо потрібну кількість світильників для приміщення з комп'ютерами при загальному рівномірному освітленні. Довжина $A=65$ м, ширина $B=19$ м, висота $H=4$ м. Висота робочої поверхні $h_p=1$ м. Для освітлення приймаємо світильник типу УПМ-11. Мінімальна освітленість лампи розжарювання з нормами $E_{\min}=100$ лк. Коефіцієнт відображення стелі $S_n=70\%$, стін $S_c=50\%$, робочої поверхні $S_p=10\%$. Напруга у мережі 220В.

1. Відстань від стелі до робочої поверхні:

$$H_0 = H - h_p = 4 - 1 = 3 \text{ м.} \quad (7.1)$$

2. Відстань від стелі до світильника:

$$h_c = 0,2 * H_0 = 0,2 * 3 = 0,6 \text{ м} \quad (7.2)$$

3. Висота підвісу світильника над поверхнею, що освітлюється:

$$h = H_0 - h_c = 3 - 0,6 = 2,4 \text{ м} \quad (7.3)$$

4. Висота підвісу світильника над підлогою:

$$H_{\Pi} = h + h_p = 2,4 + 1 = 3,4 \text{ м} \quad (7.4)$$

5. Для досягнення рівномірності освітлення приймаємо відношення

$L/h=1,5$. Тоді відстань між центром світильників:

$$L = 1,5 * p = 1,5 * 2,4 = 3,6 \text{ м} \quad (7.5)$$

6. Необхідна кількість світильників:

$$N = S/L^2 = (65 * 18,5) / 3,6^2 = 92,8 = 93 \text{ світильника} \quad (7.6)$$

Приймаємо 93 світильника (5 рядів по 19 світильників)

7. Індекс приміщення:

$$i = (A * B) / (h * (A + B)) = (65 * 18,5) / (2,4 * (65 + 18,5)) = 6,0 \quad (7.7)$$

8. За таблицею коефіцієнтів використання світлового потоку при $i=6,0$

$S_n=70\%$, $S_c=50\%$, $S_p=10\%$, для світильника типу УПМ-11 коефіцієнт використання світлового потоку - $\eta = 0,73$

9. Світловий потік однієї лампи:

$$\begin{aligned} \Phi_p &= (E_{min} * S * K_z * Z) / (N * \eta) = \\ &= (100 * 65 * 18,5 * 1,3 * 1,15) / (93 * 0,73) = 2648,02 \text{ лм} \end{aligned} \quad (7.8)$$

10. За знайденим світловим потоком з таблиці «Параметри лампи розжарювання загального призначення з розрахунковими напругами 130 та 220В» обираємо лампу потужністю 200Вт, що має світловий потік 2920 лм, найбільш близький до розрахункового.

11. Таким чином фактична освітленість E_f буде дорівнювати:

$$E_f = E_{min} * (\Phi_{л} / \Phi_p) = 100 * (2920 / 2648,02) = 110,27 \text{ лк} \quad (7.9)$$

12. Загальна потужність:

$$P_{заг} = P_{л} * N = 200 * 93 = 18600 \text{ Вт} = 18,6 \text{ кВт}. \quad (7.10)$$

7.4 Розробка заходів щодо забезпечення сприятливих умов праці при роботі з персональним комп'ютером у офісному приміщенні

Сьогодні фахівці в області ергономіки вже зрозуміли, що не можна знайти ідеальне положення, у якому можна перебувати і працювати протягом усього робочого дня. Для більшості людей комфортабельним робочим місцем повинне бути таке, котре можна пристосувати не менш чим для двох позицій, при цьому положення крісла, монітора повинні щораз відповідати виконуваний роботі і звичкам. Багато хто вважає, що для роботи на комп'ютері більше підходять вертикальне і злегка похиле положення.

Дисплей. Положення тіла звичайно відповідає напрямку погляду; дисплеї, розташовані занадто низько чи під неправильним кутом, є основними причинами сутулості. Відстань від дисплея до очей може варіюватися в залежності від характеру виконуваної роботи - 40-70 см. При роботі з текстом відстань від екрана до очей повинна лише небагато перевищувати відстань між книгою й очима і складати 40- 45 см.

Необхідно давати відпочинок очам і час від часу їх просто закривати. Не можна допускати, щоб очі увесь час були сфокусовані на одній відстані, працюючи з текстом, рекомендується установлювати великий шрифт. Дуже важливо, щоб екран монітора не мерехтів. Частота регенерації зображення повинна бути не менш 72 Гц, тому що при більш низькій частоті помітне мерехтіння, хоча люди з особливо чуттєвим зором можуть помітити його і при більш високій частоті. Їм доведеться підібрати графічну плату і монітор з частотою регенерації 85 Гц і вище.

Рекомендується встановлювати на екран монітора спеціальні скляні поляризаційні фільтри з заземленням. Необхідно уникати того, щоб термінал був звернений екраном убік вікна, оскільки інтенсивна освітленість області зору може затопити потоками світла очі і розмити зображення на сітківці. Якщо приходить сидіти поруч з вікном, то треба розташуватися під прямим

кутом до нього, причому екран дисплея повинний бути перпендикулярний шибці – цим виключаються відблиски на екрані.

Для зниження впливу низькочастотного електромагнітного випромінювання монітора на ЕЛТ рекомендується вибирати монітор, що задовольняє стандарту MPR II, чи ТСО 99. У протилежному випадку необхідно установити на екран захисний фільтр, найбільш сучасні моделі, затримують до 99 % електромагнітного випромінювання.

Крісло необхідно установити на такій висоті, щоб не почувався тиск на куприк (крісло розташоване занадто низько) на стегна (крісло розташоване занадто високо). Фахівці з ергономіки, вважали, що кут між стегнами і хребтом повинний складати 90° , однак недавно проведені дослідження показали, що більшість людей переважно сидять трохи відкинувшись.

Клавіатуру необхідно встановити так, щоб не треба було до неї тягтися і нахилитися вперед. При зміні положення тіла (наприклад, з вертикального на похиле) обов'язково треба перемінити положення клавіатури і дисплея. Може виявитися корисною регульована підставка клавіатури, завдяки якій можна без усякої напруги працювати з маніпулятором "миша". Можна поставити клавіатуру і на коліна. Руки при роботі повинні бути прямими в зап'ястях і зігнуті в ліктях приблизно під прямим кутом. Пальці також повинні бути злегка зігнуті. Удари по клавішах не повинні бути занадто сильними.

Зручна висота столу особливо важлива в тому випадку, коли на ньому розташовується клавіатура. Якщо в клавіатури немає підставки, а висоту столу не можна змінити (і він занадто високий), то треба вище підняти сидіння крісла, а під ноги підставити лавочку: чи що-небудь інше. Якщо стіл занадто низький, необхідно підкласти що-небудь під його ніжки.

Якщо при роботі часто приходиться дивитися на документи, треба установити підставку з оригіналом документа в одній площині і на одній, висоті з екраном. Якщо треба частіше дивитися на оригінал, чим на екран, необхідно повернути крісло й екран таким чином, щоб оригінал розміщався прямо перед очима, а комп'ютер ледве збоку.

Щогодини необхідно робити перерву в роботі. У цей час рекомендується робити вправи для зап'ясть. Нижче описаний можливий комплекс вправ:

- покласти руку на край столу долонею вниз. Взявшись, за пальці, іншою рукою відвести кисть назад і утримувати протягом 5 секунд.

Повторити для іншої руки;

- злегка упертися рукою в стіл і на 5 секунд напружити пальці в зап'ястя. Повторити іншою рукою;

- сильно стиснути пальці, а потім розпрямити їх. Виконання приведених рекомендацій дозволяє скоротити вплив шкідливих факторів на організм людини.

Відповідно до «Закону про охорону праці» роботодавець зобов'язаний забезпечити:

- безпеку працівників при експлуатації устаткування;
- застосовування засобів індивідуального захисту працівників;
- відповідні вимоги охорони праці, умови праці на кожному робочому місці; дотримання режиму праці і відпочинку працівників;
- навчання безпечним методам і прийомам виконання робіт;
- інструктаж з охорони праці;
- організацію контролю за станом умов праці на робочих місцях;
- проведення атестації робочих місць за умовами праці;
- інформування працівників про умови й охорону праці на робочих місцях, про існуючий ризик ушкодження здоров'я і компенсаціях при ушкодженні та засобах індивідуального захисту.

Вимоги до освітлення. У приміщеннях туристичної агенції, де експлуатуються комп'ютери, штучне освітлення повинне бути загальним і рівномірним. Однак якщо співробітники переважно працюють з документами, то допускається застосувати комбіноване освітлення: крім загального установлюються світильники місцевого освітлення, які не повинні створювати відблисків на поверхні екрана і збільшувати його освітленість

більш 300 лк. Освітленість поверхні столу в зоні розміщення робочого документа повинна становити 300-500 лк.

Джерела освітлення варто встановлювати таким чином, щоб вони не осліплювали, при цьому яскравість світних поверхонь (вікна, світильники та ін.), які розташовуються в полі зору, повинна бути не більш 200 кд/м^2 .

В якості джерела світла при штучному освітленні повинні застосовуватися переважно люмінесцентні лампи типу ЛБ. При пристрої відбитого освітлення допускається застосування металогалогенних ламп потужністю до 250 Вт, а у світильниках місцевого освітлення – ламп накаливання.

Для забезпечення нормованих значень освітленості в приміщеннях туристичної агенції необхідно не рідше двох разів на рік чистити скло, віконні рами і світильники та вчасно замінювати перегорілі лампи.

Робочі місця повинні розташовуватися таким чином, щоб природне світло падало збоку, переважно ліворуч.

Для внутрішньої обробки приміщень повинні використовуватися дифузно-відбиваючі матеріали, з коефіцієнтом відбиття від стелі – 0,7-0,8; для стін – 0,5-0,6; для підлоги – 0,3-0,5. Полімерні матеріали для внутрішньої обробки повинні бути дозволені для застосування органами й установами Держсанепіднагляду.

Поверхня підлоги в приміщеннях повинна бути рівною, без вибоїн, неслизькою, зручною для очистки і вологого прибирання, мати антистатичні властивості.

Вимоги до кондиціонування. У приміщеннях туристичної агенції, у яких установлені комп'ютери, мікроклімат повинен відповідати наступним санітарним нормам:

- температура повітря в теплий період року – не більш 23-25 С, у холодний – 22-24 С;
- відносна вологість повітря – 40-60%;
- швидкість руху повітря – 0,1 м/с.

Для підвищення вологості повітря в приміщеннях варто застосовувати зволожувачі повітря, щодня заправляти їх дистильованою або кип'яченою водою. [43]

Вимоги до розміщення, оснащення й організації робочих місць. Відстань між робочими столами з моніторами (у напрямку тилу поверхні одного монітора та екрана іншого) повинне бути не менш 2 м, а між бічними поверхнями моніторів – не менш 1,2 м.

Віконні отвори повинні бути обладнані регульованими жалюзі, завісами, зовнішніми навісами та ін.

Бажано, щоб висоту робочої поверхні столу можна було регулювати в межах 680-800 мм, а при відсутності такої можливості вона повинна дорівнювати 725 мм. Модульними розмірами робочої поверхні комп'ютерного столу, на базі яких розраховують конструктивні розміри, варто вважати: ширину 800, 1000, 1200 і 1400 мм; глибину 800 і 1000 мм.

Екран монітора повинен знаходитись від очей користувача на оптимальній відстані 600-700 мм, але не ближче 500 мм з урахуванням розмірів алфавітно-цифрових знаків і символів.

Вимоги до безпеки під час експлуатації й обслуговування ЕОМ. У приміщенні з комп'ютерами повинно проводитися щоденне вологе прибирання. Приміщеннях туристичної агенції повинні оснащуватися аптечкою першої допомоги та вуглекислотними вогнегасниками.

Режими праці і відпочинку. Режими праці і відпочинку при роботі на комп'ютерах залежать від виду і категорії трудової діяльності.

При восьмигодинній робочій зміні і роботі на комп'ютері регламентовані перерви варто встановлювати:

- для I категорії робіт – через 2 години від початку робочої зміни і через 2 години після обідньої перерви тривалістю по 15 хв;
- для II категорії робіт – через 2 години від початку робочої зміни і через 1,5 - 2 години після обідньої перерви тривалістю по 15 хв. або через кожную годину роботи тривалістю по 10 хв.

Під час регламентованих перерв із метою зниження нервово-емоційної напруги, зменшення стомлення очей, усунення гіподинаміки і гіпокінезії доцільно виконувати комплекси вправ, викладених у санітарних нормах.

7.5 Висновки до розділу 7

Метою ОП є забезпечення безпечних, незагрозливих та придатних для праці умов.

Під час роботи на ПК робітники відділення мають справу з дією таких небезпечних факторів: підвищена температура, погане освітлення робочого місця, електричний струм, інтелектуальне або емоційне перевантаження, перенапруження очей, монотонна праця.

Робочий офіс фірми займає 3-поверхове приміщення у м. Миколаїв.

Система пожежогасіння у офісному приміщенні та шляхи евакуації з приміщення повністю відповідають нормам. Захисна установка заземлення може вважатись прийнятною та допустимою згідно з ПЕУ-86.

Отже, аналіз шкідливих та небезпечних чинників у компанії «Template Monster» показав, що загальні робочі умови, мікроклімат, штучне освітлення, заземлення та система пожежної безпеки офісі відповідає нормам та стандартам.

8 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

8.1 Основні положення охорони навколишнього середовища

Охорона навколишнього середовища являє собою форму відносин між суспільством і природою. Вона здійснюється різними засобами: економічними, правовими, науково-технічними, санітарно-гігієнічними, біологічними та іншими.

В загальному випадку проблема охорони навколишнього середовища зводиться до вирішення двох завдань:

- організації раціонального природокористування;
- забезпечення чистоти природних (екологічних) систем.

При здійсненні різних видів економічної діяльності суб'єкти господарювання використовують різноманітні природні ресурси: землю, воду, корисні копалини тощо. Проте ресурси ці обмежені. Обмеженість природних ресурсів була і залишається головною і дуже жорсткою умовою, що накладається на розвиток економіки і відповідно зростання суспільного добробуту.

Наслідком обмеженості природних ресурсів є конкуренція за їх застосування, тобто суперництво між альтернативними цілями використання ресурсів. Адже майже всі ресурси можуть використовуватися для задоволення найрізноманітніших потреб. Наприклад, нафта може служити сировиною для одержання палива, виробництва синтетичних волокон, пластмас, лакофарбових виробів, побутової хімії тощо. І всі ці альтернативні цілі конкурують за використання сирової нафти, обсяги якої, як відомо, обмежені.

Раціональне природокористування означає розробку та здійснення концепції і конкретних заходів щодо раціонального використання і

відтворення природних ресурсів, гармонічну взаємодію суспільства і природи, людини і навколишнього природного середовища.

Завдання організації раціонального природокористування вирішується шляхом:

- оптимального розподілу ресурсів між різними господарськими цілями;
- використання технологій, що зберігають ресурси;
- проведення заходів щодо поповнення природних ресурсів.

Економічна діяльність у всіх її проявах здійснює забруднення навколишнього середовища. У процесі цієї діяльності забруднюються і стають дефіцитними ресурси повітря, води, територій, що здавалися нескінченними. Нині рівень забруднення досяг загрозливих розмірів, набувши по суті кризового характеру.

Однією з причин забруднення навколишнього середовища є збільшення обсягу відходів та викидів. До них відносять: не використані у виробництві матеріали, що не підлягають подальшій переробці, або продукти, що відслужили свій термін споживання, різні пакувальні матеріали, всілякі відвали та терикони породи тощо.

Величезна кількість відходів є результатом значного збільшення обсягів виробництва. Проблема утилізації відходів стала сьогодні проблемою глобального масштабу.

8.2 Забруднення навколишнього середовища підприємством.

Заходи щодо зменшення забруднення навколишнього середовища

Персональні комп'ютери, ноутбуки та інша інформаційна техніка, як відомо широко використовується в галузі наукових досліджень, промисловості, а також у повсякденному домашньому користуванні. Але будь-яка техніка стрімко застаріває, їй на зміну приходять нові, більш

потужні, більш сучасні ПК та оргтехніка. Поступово виникає проблема, що робити зі старою технікою, морально застарілою або з тих чи інших причин, що вийшла з ладу, яка захищає підсобні приміщення та склади.

Утилізація оргтехніки та комп'ютерів це процес, який проводиться в кілька етапів. Найперше дію це списання обладнання безпосередньо з підприємства. Етап другий це розбір техніки і сортування отриманих матеріалів. Якщо деталі здатні служити вихідною сировиною, наприклад, кінескоп, деталі, в складі яких є дорогоцінні метали, то їх відправляють на очищення.

Як нам відомо до складу комп'ютера входить безліч металів таких як золото, срібло, алюміній, мідь та інших. Ще етапом по утилізації персональних комп'ютерів можна досягнути завдяки вторинній переробці.

Сутність даного процесу полягає в тому, що можна витягти з цієї сировини частку корисних та рідких матеріалів таких як іридію, міді та інших. Цей процес відтворити набагато легше ніж наприклад видобути тонну міді, яка міститься в тисячотонних гірських породах.

Одне з нововведень для утилізації друкованих плат придумали Співробітники з Національної фізичної лабораторії Великобританії, продемонстрували можливість спеціального розчину який розчинюють у гарячій воді. Дія якого зумовлює відшарування електронних компонентів.

Таким чином 90% компонентів нових друкованих плат можна використовувати знову, тоді як у випадку звичайним методам - тільки 2%.

На прикладі сучасного стану комп'ютерних технологій, їх активного раз-розвитку в контексті націленості на екологічну заклопотаність, можна проаналізувати, як реалізуються на практиці принципи екологічної етики.

Мотивація до розробки та впровадження енергозберігаючих комп'ютерних технологій спирається на прагматичний підрахунок витрачених коштів на оплату споживаної електроенергії. Як основний аргумент наводиться масштаб і кількість енергоємних установок, починаючи з персонального комп'ютера в будинку, закінчуючи великими дата-центрами,

що споживають велику кількість електроенергії , при цьому виділяється велика кількість теплоти, на зниження якої також використовуються додаткові енергоресурси.

Як відомо, пластмаса, яка використовується у виробництві комп'ютерів, не «переробляється» природою відразу. Викинутий на смітник старий системний блок може пролежати до повного розкладання і сто, і двісті років. Скло електронно-променевої трубки дисплея протримається в сирій землі близько тисячоліття , а в більш сприятливих умовах і довше, до цілого мільйона років!

Загальна інформованість про проблеми екологічної загрози, пропаганда ідей соціальної відповідальності бізнесу та індивідуальної практики малих справ, маркетингові проекти , що спираються на потреби сучасного суспільства, дозволяють створити умови для поступового просування ідей і принципів екологічної етики.

Очевидно, що ІТ- галузь повинна знизити вплив швидкого зростання інформації на навколишнє середовище, забезпечивши більш високі рівні ефективності.

Три основні технологічні ініціативи допомагають вирішити цю задачу:

- Віртуалізація та консолідація
- Управління життєвим циклом інформації
- Дедуплікація даних

Віртуалізація та консолідація необхідні для скорочення енергоспоживання в ЦОД, в яких ІТ-менеджери встановлюють все більше систем для підвищення продуктивності, надмірності та доступності, не думаючи про ефективність живлення або охолодження. Віртуалізація вирішує проблему неефективності, відокремлюючи ПЗ від базового обладнання, що забезпечує роботу декількох ОС і додатків на одному комп'ютері. У свою чергу, більш високий коефіцієнт використання серверів і сховищ означає розгортання меншої кількості машин, які споживають менше електроенергії для живлення та охолодження.

Управління життєвим циклом інформації (ILM) ґрунтується на тій передумові, що цінність інформації змінюється з часом. ILM використовує автоматичний аналіз для зберігання інформації в найбільш підходящому енергетично ефективному сховищі на всіх етапах її життєвого циклу. Наприклад, оперативні дані і дані, важливі для роботи бізнесу, потребують систем з максимальною надійністю і продуктивністю, які вимагають більше ресурсів і потужності. Коли важливість інформації знижується, ILM переносить її в сховище з меншим енергоспоживанням.

Дедуплікація даних значно знижує обсяг зберігаються резервних копій, обумовлений тим, що користувачі зберігають по кілька копій і варіацій одного файлу в різних точках мережі. Резервне копіювання вихідного файлу на центральний сервер виконується тільки один раз. Зміни файлу відправляються на сервер у вигляді нових і унікальних сегментів даних, пов'язаних з оригіналом. Резервне копіювання виконується тільки для цих нових сегментів. [44]

Створення екологічної стійкості - масштабне підприємство. Воно вимагатиме глобального співробітництва і спільної роботи між найбільшими і впливовими зацікавленими сторонами - ООН, державними та приватними установами, компаніями, університетами та приватними особами. Воно вимагатиме системного погляду на вироблення електроенергії, енергоспоживання і зміна клімату, а також комбінації стимулів, стандартів, грантів, безперервних досліджень, уяви та інновацій, що сприяють глобальним змінам, які необхідні якомога швидше.

8.3 Висновки до розділу 8

Охорона навколишнього середовища є формою відносин між природою та суспільством. Існують різні засоби, через які вона може здійснюватись.

Серед них: економічні, правові, гігієнічні, науково-технічні, біологічні та інші.

Дві основні проблеми, які вирішуються у процесі охорони навколишнього середовища – це забезпечення чистоти, її підтримання та організація раціонального природокористування.

Будь-яка економічна діяльність забруднює природне середовище. Як результат – дефіцит та забрудненні повітряні ресурси, території, вода.

За порушення у сфері охорони природи, незаконне використання природних ресурсів законодавством України передбачена адміністративна (статті 52 - 92 Кодексу про адміністративні правопорушення) та кримінальна (статті 236 - 254 Кримінального кодексу) відповідальність.

Утилізація– це обов'язкова процедура для підприємств і організацій різної форми власності. Неналежне виконання цієї процедури призводить до податкової та адміністративної відповідальності.

ВИСНОВКИ

У сучасні автомобілі вбудовані електронні прилади, які відстежують і контролюють роботу авто. Якщо виникає проблема, не завжди відразу виходить виявити її, тим більше - визначити причину. Тут потрібна професійна автодіагностика з використанням певного спеціального і універсального обладнання. Іноді порушення в роботі агрегатів неможливо виявити через несправні індикаторів і датчиків. Вони відображають показники, які не відповідають реальній картині. Ось чому потрібна систематична діагностика автомобіля.

Якість програмного забезпечення являє собою один з найцікавіших, але у той же час найскладніших, атрибутів, який впливає на досвід використання ПЗ.

Актуальність проблеми отримання ефективної системи оцінювання якості ПЗ важливим завданням, що вимагає удосконалення існуючих методів. Адже саме якість розробки програм може стати відправною точкою для успіху або невдачі проекту на ранньому етапі розробки.

Основною метою магістерської роботи було удосконалення моделі якості програмного забезпечення діагностики двигунів та розробка ПЗ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ГОСТ 19.102–77. Единая система программной документации. Стадии разработки // Единая система программной документации. – М.: Издательство стандартов, 1985. – С. 34–36.
2. Орлов, С.А. Технологии разработки программного обеспечения / С.А. Орлов. – СПб: Питер, 2002. – 464 с.
3. Зингер, К. Методы проектирования программных систем / К. Зингер. – М.: Мир, 1985. – 328 с.
4. Калянов, Г.Н. CASE структурный системный анализ / Г.Н. Калянов. – М.: ЛОРИ, 1996. – 242 с.
5. Требования и спецификации в разработке программ. Сборник статей./ Под редакцией В.Н. Агафонова. / . – М.: Мир, 1984. – 344 с.
6. Кинг, Д. Создание эффективного программного обеспечения / Д. Кинг. – М.: Мир, 1991. – 288 с.
7. Якобсон, А., Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г. Буч, Дж. Рамбо . – СПб: Питер, 2002. – 496 с.
8. Буч, Г. Язык UML. Руководство пользователя: пер. с англ / Г. Буч, Дж. Рамбо, А. Джекобсон. – М.: ДМК, 2000. – 432 с.
9. Майерс, Г. Искусство тестирования программ / Г. Майерс. – М.: Финансы и статистика, 1982. – 176 с.
10. ГОСТ 19.201–78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению // Единая система программной документации. – М.: Издательство стандартов, 1985. – С. 63–66.
11. Briand, L.C. Resource Estimation in Software Engineering [Text] / L.C. Briand, I. Wiczorek // Encyclopedia of Software Engineering. – John Wiley & Sons, Inc., 2002. –Р. 83.
12. Магнус, Я.Р. Эконометрика. Начальный курс: Учеб. – 6-е изд., перераб. и доп. [Текст] / Я.Р. Магнус, П.К. Катышев, А.А. Пересецкий. – М.: Дело, 2004. – С. 576.

13. Гмурман, В.Е. Теория вероятностей и математическая статистика [Текст]: Учеб. пособие. – 12-е изд., перераб. / В.Е. Гмурман. – М.: Высшее образование, 2010. –С. 479.

14. International Organization for Standardization/International Electrotechnical Commission, 2001. ISO/IEC 9126-1 Standard, Software Engineering, Product Quality, Part 1: Quality Model, Tokyo.

15. Демиденко, Е.З. Линейная и нелинейная регрессии [Текст] / Е.З. Демиденко. – М.: Финансы и статистика, 1981. – С. 302.

16. Bates, D.M. Nonlinear Regression Analysis and Its Applications [Text] / D.M. Bates, D.G. Watts, – Wiley, 1988. – P. 384.

17. Pardoe, I. Applied regression modeling [Text] / I.Pardoe. – Wiley, 2012. – P. 325.

18. Уніфікована мова моделювання - URL: <https://bibliofond.ru/view.aspx?id=446563#1> (дата звернення: 19.09.2019)

19. Діаграма варіантів використання - URL: https://studopedia.ru/19_284009_diagrama-variantiv-vikoristannya-USE-case-diagram.html (дата звернення: 19.09.2019)

20. Діаграма активності - URL: <http://khp-iip.mipk.kharkiv.edu/library/case/leon/gl7/gl7.html> (дата звернення: 19.09.2019)

21. Діаграма станів - URL: <http://moodle.ipk.kpi.ua/moodle/mod/resource/view.php?id=2808710> (дата звернення: 20.09.2019)

22. Джеф Раскин. Интерфейс: новые направления в проектировании компьютерных систем [Текст] / Дж. Раскин. – К.: Символ-Плюс, 2015. – С. 272.)

23. Діаграма класів - URL: <http://moodle.ipk.kpi.ua/moodle/mod/resource/view.php?id=28088> – (дата звернення: 21.09.2019)

24. Діаграма послідовності - URL:
<http://moodle.ipk.kpi.ua/moodle/mod/resource/view.php?r=16538> (дата
звернення: 21.09.2019)
25. Нотон Патрик, Шилдт Герберт. Полный справочник по Java.: Пер. с
англ.: - К.: Диалектика, 1997
26. Герберт Шилдт. Java. Полное руководство. 10-е издание [Текст] /
Г. Шилдт. – К.: Диалектика, 2019.– С. 730.
27. Поняття, мета і завдання охорони праці – URL:
http://ncpn.net.ua/oxrana_truda.html (дата звернення: 11.09.2019)
28. ДСТУ Б В.2.2-6-97 (ГОСТ 24940-96) Здания и сооружения. Методы
измерения освещенности [Текст]. – Введ. 1997-09-15 – М.: Стандартиформ,
1998. – С. 4.
29. ГОСТ 12.1.005-88. Система стандартов безопасности труда. Общие
санитарно-гигиенические требования к воздуху рабочей зоны [Текст]. – Введ.
1989-01-01 – М.: Стандартиформ, 2000. – С. 4.
30. Охрана навколишнього середовища – URL:
http://pidruchniki.com/13351207/ekonomika/ohorona_navkolishnogo_seredovisha.
(дата звернення: 11.09.2019)

Додаток А

Технічне завдання

Вступ

Назва розроблюваного проекту: «Програмне забезпечення діагностики двигунів». Область застосування – підприємство, яке займається ремонтом авто транспорту.

1. Підстави для розробки

Підставою для розробки є завдання на магістерську роботу, видане кафедрою програмного забезпечення автоматизованих систем науково-навчального інституту комп'ютерних наук проектами Національного університету кораблебудування ім. адмірала Макарова.

2. Призначення розробки

2.1 Експлуатаційне призначення

Експлуатаційним призначенням програмного забезпечення є автоматизація діагностики двигунів.

2.2 Функціональне призначення програми

Функціональним призначенням програмного забезпечення є автоматизація процесів:

- регулювання показників двигуна;
- зчитування помилок двигуна при експлуатації;
- розрахунку параметрів;

3.1.2 Вимоги до організації вхідних та вихідних даних

Введення даних здійснюється за допомогою пристроїв введення даних (клавіатура або миша).

Виведення даних здійснюється за допомогою пристрою виведення даних – монітору комп'ютера.

Вхідні дані:

- Данні з ЕБУ;
- параметри датчиків двигуна.

Вихідні дані:

- Розраховані параметри Вимоги до надійності

Програмний продукт повинен нормально функціонувати при безперебійній роботі ПК. При виникненні збоїв в роботі, відновлення нормальної роботи повинне проводитися після перезавантаження програми.

Відмови програми можливі внаслідок некоректних дій користувача при взаємодії з програмою. Щоб зменшити кількість відмов програми за вказаною вище причиною слід забезпечити зрозумілість програми, через надання підказок кінцевому користувачеві у разі невірних дій.

3.2 Вимоги до умов експлуатації

Необхідний рівень підготовки користувачів: мінімальні навички в користуванні комп'ютером. Комп'ютер призначений для роботи в закритому опалювальному приміщенні при наступних умовах навколишнього середовища: температура повітря від $+10^{\circ}\text{C}$ до $+30^{\circ}\text{C}$; відносна вологість повітря не більше 80; запиленість повітря не більше $0,75 \text{ мг/м}^2$.

3.3 Вимоги до складу та параметрів технічних засобів

Система користувача повинна задовольняти вказаним мінімальним вимогам для коректного запуску програми:

- CPU: Intel Pentium 4405Y (1.2 ГГц);
- RAM: 1 GB;
- 500 Mb вільного місця на диску.

3.4 Вимоги до інформаційної та програмної сумісності

Для повноцінного функціонування системи необхідно використовувати: ОС Windows версії не нижче 7.

3.5 Вимоги до маркування та пакування

Програмне забезпечення діагностики двигунів, буде упаковано в електронний архів і мати найменування «motor_test.rar».

3.6 Вимоги до транспортування та зберігання

Вимоги до транспортування не висовуються у зв'язку з відсутністю фізичних носіїв.

3. Вимоги до програмної документації

Програмна документація повинна містити:

- технічне завдання.
- опис програми
- текст програми
- інструкція користувача
- програма і методика випробувань ПЗ

4. Техніко-економічні показники

У даному розділі розраховуються такі показники:

- вартість програм;
- амортизаційні відрахування на устаткування;
- витрати на основні і допоміжні матеріали;
- експлуатаційні витрати;
- сумарні витрати на створення й експлуатація програми за перший рік;
- річна економічна ефективність;
- строк окупності ПЗ.

5. Стадії та етапи розробки

Стадії та етапи розробки представлені нижче в таблиці А.1.

Таблиця А.1 — Стадії та етапи розробки проекту

Стадії розробки	Етапи робіт	Термін виконання робіт	
		Початок етапу	Кінець етапу
1. Технічне завдання	1.1 Обґрунтування необхідності розробки програми	22.04.20	26.04.20
	1.2 Розробка технічного завдання	29.04.20	10.05.20
	1.3 Затвердження технічного завдання	13.05.20	16.05.20
2. Ескізний проект	2.1 Розробка ескізного проекту	20.05.20	05.06.20
	2.2 Затвердження ескізного проекту	07.06.20	11.06.20
3. Технічний проект	3.1 Розробка технічного проекту	02.09.20	20.09.20
	3.2 Затвердження технічного проекту	23.09.20	27.09.20
4. Робочий проект	4.1 Розробка робочого проекту	30.09.20	11.10.20
	4.2 Затвердження робочого проекту	15.10.20	18.10.20
	4.3 Розробка документації	21.10.20	1.11.20

6. Порядок контролю та прийому

Контроль за аналізом та проектуванням кожної окремої частини програмного забезпечення для автоматизації обробки інформації для оцінювання часу виконання веб-проектів мовою Liquid, відбувається на кожному етапі з урахуванням вимог, визначених у технічному завданні. Кожна стадія розробки повинна бути представлена в зазначені строки та узгоджена із замовником.

Прийом готового програного проекту документують за допомогою протоколу проведення випробувань. У разі знаходження помилок під час прийому програмного виробу складається акт про знайдені помилки, який підписується представниками замовника і розробника і затверджується керівниками організації - замовника та організації - розробника. Розробник повинен протягом не більше ніж 2 тижнів виправити зазначені помилки, і оповістити замовника про повторне проведення перевірки.

Додаток Б

Текст програми

```

package Solution;

import java.util.Arrays;
import java.util.Collection;
import java.util.HashSet;
import java.util.Scanner;
import java.util.TreeSet;

public class StatTable {

    public static void main(String[] args) throws Exception {
        countWords(new BinarySearchTree<String>(), allFileNames[0]);
        for(String s : allFileNames) {
            System.out.println(s);
            countWordsOurUnsortedSet(s);
            countWordsOurBinarySearchTree(s);
            countWordsOurHash(s);
            countWordsCollection(new TreeSet<String>(), s);
            int[] result = countWordsCollection(new HashSet<String>(), s);
            System.out.println(result[0] + " total words.");
            System.out.println(result[1] + " distinct words.");
            System.out.println();
        }
    }

    public static int[] countCollection(Collection<String> c, String
fileName) throws Exception{
        c.clear();
        Scanner fileScanner = new Scanner(new File(fileName));
        Stopwatch st = new Stopwatch();
        st.start();
        int total = 0;
        while(fileScanner.hasNext()){
            c.add(fileScanner.next());
            total++;
        }
        st.stop();
        System.out.println("Time for " + c.getClass() + " : \n" + st);
        System.out.println(c.size() + " distinct words");
        System.out.println(total + " total words including duplicates: ");
        assert total >= c.size();
        System.out.println();
        return new int[]{total, c.size()};
    }

    public static int[] countWordsOurHash(String fileName) throws Exception {
        Scanner fileScanner = new Scanner(new File(fileName));
        Stopwatch st = new Stopwatch();
        UnsortedHashSet<String> c = new UnsortedHashSet<String>();
        st.start();
        int total = 0;
        while(fileScanner.hasNext()) {
            c.add(fileScanner.next());
            total++;
        }
    }

```

```

        st.stop();
        System.out.println("Time for our hashtable (closed address hashing):
\n" + st);
        System.out.println(c.size() + " distinct words");
        System.out.println(total + " total words including duplicates: ");
        assert total >= c.size();
        System.out.println();
        return new int[]{total, c.size()};
    }

    public static int[] countWordsOurUnsortedSet(String fileName) throws
Exception {
        Scanner fileScanner = new Scanner(new File(fileName));
        Stopwatch st = new Stopwatch();
        UnsortedSet<String> c = new UnsortedSet<String>();
        st.start();
        int total = 0;
        while(fileScanner.hasNext()){
            c.add(fileScanner.next());
            total++;
        }
        st.stop();
        System.out.println("Time for our unsorted set based on ArrayList: \n"
+ st);
        System.out.println(c.size() + " distinct words");
        System.out.println(total + " total words including duplicates: ");
        assert total >= c.size();
        System.out.println();
        return new int[]{total, c.size()};
    }

    public static int[] countWordsOurBinarySearchTree(String fileName) throws
Exception {
        Scanner fileScanner = new Scanner(new File(fileName));
        Stopwatch st = new Stopwatch();
        BinarySearchTree<String> c = new BinarySearchTree<String>();
        st.start();
        int total = 0;
        while(fileScanner.hasNext()){
            c.add(fileScanner.next());
            total++;
        }
        st.stop();
        System.out.println("Time for our binary search tree: \n" + st);
        System.out.println(c.size() + " distinct words");
        System.out.println(total + " total words including duplicates: ");
        assert total >= c.size();
        System.out.println();
        return new int[]{total, c.size()};
    }

    public static int[] countWords(Object c, String fileName) throws
Exception {
        Scanner fileScanner = new Scanner(new File(fileName));
        Stopwatch st = new Stopwatch();
        System.out.println(Arrays.toString(c.getClass().getMethods()));
        Method addMethod = c.getClass().getMethod("add", Object.class);
        st.start();
        int total = 0;
        while(fileScanner.hasNext()){
            addMethod.invoke(c, fileScanner.next());
            total++;
        }
        st.stop();
        System.out.println("Time for " + c.getClass() + ": " + st);
    }

```



```

        Method sizeMethod = c.getClass().getMethod("size");
        int distinctWords = (Integer) sizeMethod.invoke(c);
        System.out.println(distinctWords + " distinct words");
        System.out.println(total + " total words including duplicates: ");
        System.out.println();
        return new int[]{total, distinctWords};
    }
}

void hi_square(std::vector<double>&a) {
    std::vector<int> counter;
    for (int i = 0; i < 10; ++i) {
        counter.push_back(std::count_if(a.begin(), a.end(), [&i](double
val) {
            return (val < i * 0.1 + 0.1 && val > i * 0.1);
        }));
    }
    double hi = 0;

    for (auto item : counter) {
        hi += std::pow(item - arraySize * 0.1, 2) / (arraySize * 0.1);
    }
    return hi;
}

public:
    enum RegressionType{Regres, Classif};

    Regression(const DataSet&, const char*);
    ~Regression();

    double gradientdescent(mat, const mat, const double, const unsigned int);

    mat theta(void) const;
    void init_theta(void);
    void printTheta(void) const;

    string regressionType(void) const;
    void set_regressionType(const string&);

    double alpha(void) const;
    void set_alpha(const double);

    double lamda(void) const;
    void set_lamda(const double);

    virtual vec h_Theta(vec) const = 0;
    virtual double cost(mat&, const mat&) const = 0;
    virtual mat derivative(const mat&, const mat&) const = 0;

protected:
    mat d_Theta;

    const DataSet& d_dset;
    RegressionType d_reg_type;

    double d_alpha;
    double d_lamda;

    fstream d_lamdaCostGraph;
};

```

Додаток В

Опис програми

Програмне забезпечення діагностики двигунів, яке розроблялась у рамках магістерської роботи, спрямована на використання на підприємствах, які займаються ремонтом автотранспорту.

Функціональним призначенням програмного забезпечення є автоматизація процесів:

- регулювання показників двигуна;
- зчитування помилок двигуна при експлуатації;
- розрахунку параметрів;

Для розробки програмного забезпечення було обрано мову програмування Java та середовище розробки NetBeans.

При розробці додатку значну увагу було приділено інтерфейсу користувача. Важливим завданням було створити його якомога простим у використанні і в той самий час багатофункціональним і зручним.

Додаток Г

Інструкція користувача

Програмне забезпечення автоматизованої обробки інформації для оцінювання часу виконання веб-проектів мовою liquid, яка розроблялась у рамках магістерської роботи, запакована в архів з назвою MotorTest.zip.

Після розпакування архіву заходимо в папку та відкриваємо програму MotorTest. Перед нами з'являється початкове вікно програми (рисунок Г.1):

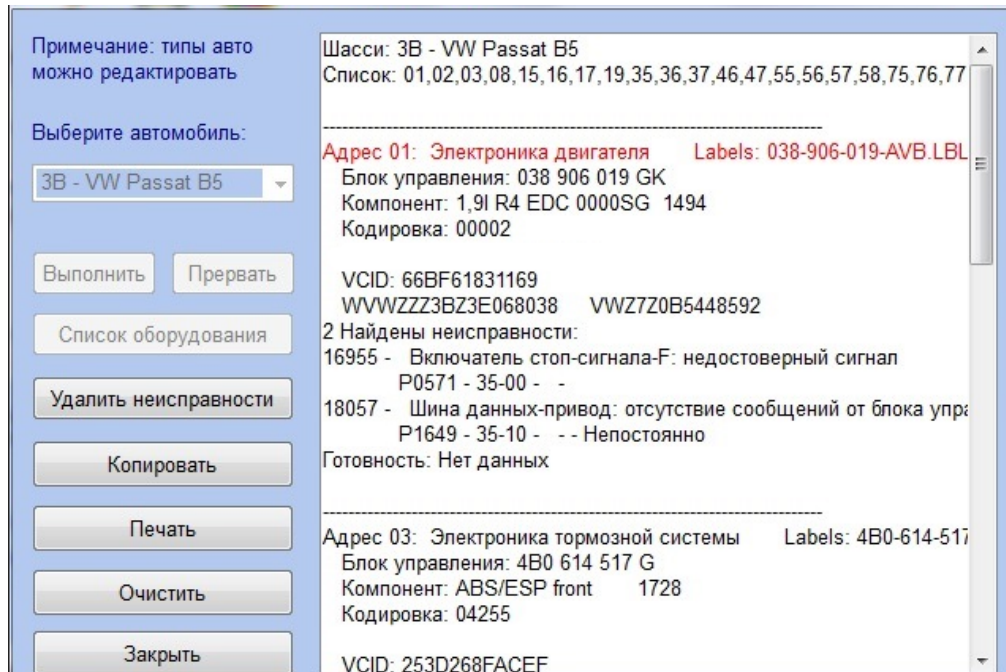


Рисунок Г.1 - Початкове вікно програми

Опис функцій та інформаційних блоків, наявних на початковому вікні програми наведено на рисунку Г.1:

1. Кнопка «Удалить неисправности»
2. Кнопка «Копировать»
3. Кнопка «печать»
4. Кнопка «Очистить»

Додаток Д

Програма і методика випробувань програмного забезпечення

1 Об'єкт випробування

Об'єктом випробування є програмне забезпечення діагностики двигунів, розроблене в рамках даної дипломної роботи.

2 Мета випробування

Метою проведення випробування є перевірка працездатності даного програмного продукту, перевірка відповідності характеристик та вимог розробленої програми, викладених у технічному завданні, приведення прикладу роботи та отримання відповідних навичок.

3 Вимоги до додатка

При проведенні випробувань функціональні характеристики (можливості) програми підлягають перевірці на відповідність вимогам, викладеним у пункті «Вимоги до функціональних характеристик» технічного завдання.

4 Вимоги до програмної документації

4.1 Склад програмної документації, пропонованої на випробування

Програмна документація повинна включати в себе наступні документи:

- 1) Технічне завдання;
- 2) Текст програми;
- 3) Опис програми;
- 4) Інструкція користувача;
- 5) Програма та методика випробувань.

4.2 Спеціальні вимоги

Спеціальні вимоги до програмної документації не висуваються.

5 Засоби і порядок випробувань

5.1 Технічні засоби, що використовуються під час випробувань

Склад технічних засобів, що використовувалися під час випробувань:

- CPU: Intel Pentium 4405Y (1.2 ГГц);
- RAM: 1 GB;
- Жорсткий диск 70Гб RAID 1;
- Здатність екрану – 1920×1080 пікселів;
- Клавіатура 101/102-х клавішна рус / лат;
- Маніпулятор «миша».

5.2 Програмні засоби, що використовуються під час випробувань

- Кабель OBD-II

Склад програмних засобів, що використовувалися під час випробувань:

Операційні системи:

- Windows 7.

5.3 Порядок проведення випробувань

Порядок проведення випробувань складається з перевірки вимог до функціональних характеристик програмного продукту та перевірки вимог до програмної документації.

6 Методи випробувань

6.1 Методика проведення перевірки вимог до програмної документації

Перевірка дотримання вимог програмної документації на програмний продукт виконується візуально представником служби, відповідальної за експлуатацію. У ході перевірки зіставляється склад і комплектність

програмної документації, представленої розробником, з переліком програмної документації, наведеним у пункті «Склад програмної документації, пропонованої на випробування» цього документа.

Перевірка вважається завершеною у випадку відповідності складу та комплектності програмної документації, представленої розробником, переліком програмної документації, наведеному у зазначеному вище пункті.

6.2 Методика проведення перевірки вимог до функціональних характеристик програмного продукту

Перевірка працездатності програми виконується згідно з пунктом «Вимоги до функціональних характеристик» технічного завдання.

Перевірка вважається завершеною у разі відповідності складу і послідовності дій, при виконанні даної перевірки, вказаною вище пункту технічного завдання.

В процесі тестування була перевірена функціональність програмне забезпечення діагностики двигунів. У таблиці, що наведено нижче, вказано перелік випробувань основних функціональних можливостей.

Таблиця Д1 - Методика випробування

	Дія	Очікуваний результат	Результат перевірки	Зауваження
	2	3	4	5
1	Випробування на завантаження з вимкненим запалюванням	Виводиться повідомлення про помилку	Виконано	
2	Випробування на завантаження з ввімкненим запалюванням	З'являється панель параметрів	Виконано	

Продовження таблиці Д1

	2	3	4	5
3	Випробування на розрахунок параметрів після завантаження	У відповідних текстових блоках з'явилися розраховані параметри	Виконано	
4	Випробування на розрахунок параметрів	У результаті з'явилося відповідне системне повідомлення, про необхідність ввести коректні параметри	Виконано	